Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática
2013

Paulo Lopes Simões

Plataforma para Aquisição de Sinais de Stress e Registo de Ocorrências
Platform to Acquire Stress Signals and Register Occurrences

Universidade de Aveiro Departamento de
Electrónica, Telecomunicações e Informática
2013

Paulo Lopes Simões

Plataforma para Aquisição de Sinais de Stress e Registo de Ocorrências
Platform to Acquire Stress Signals and Register Occurrences

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requesitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica do Doutor José Alberto Gouveia Fonseca, Professor Associado do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro e da Mestre Maria Margarida Urbano, Professora da Escola Superior de Tecnologia e Gestão de Águeda.

o júri / the jury

presidente / president	**Prof. Doutor Rui Manuel Escadas Ramos Martins** Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro
vogais / examiners committee	**Prof. Doutor José Alberto Gouveia Fonseca** Professor Associado do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro (orientador)
	Mestre Maria Margarida Carreira Pires Urbano Professora Adjunta da Escola Superior de Tecnologia e Gestão de Águeda (co-orientadora)
	Prof. Doutor Viriato António Pereira Marinho Marques Professor Coordenador, ISEC Instituto Superior de Engenharia de Coimbra

agradecimentos / acknowledgements

É com muito gosto que aproveito esta oportunidade para agradecer a todos os que me ajudaram durante todo o meu percurso académico.

Um agradecimento especial para os meus orientadores José Alberto Fonseca e Margarida Urbano pelo empenho, disponibilidade e excelente carácter humano mostrado ao longo de todo este trabalho realizado.

Um agradecimento muito especial para a minha família, nomeadamente os meus pais e o meu irmão, pelo esforço e motivação que depositaram em mim para que pudesse seguir este percurso académico.

Agradeço também a Alexandra Sîrbu por todo o seu apoio e acompanhamento. Um agradecimento também a todos os meus grandes amigos.

Quero também agradecer aos meus colegas do laboratório 319 do DETI pelo bom ambiente criado, importante para a elaboração deste trabalho. Um agradecimento aos meus colegas pela entreajuda ao longo do curso.

Um agradecimento especial ao meu amigo Helder Lopes Francisco que teve a gentileza de elaborar a imagem de fundo e ícone da applicação Android desenvolvida neste projecto, dando assim um aspecto mais profissional.

Para finalizar, quero agradecer ao ISR (Institute of Systems and Robotics) da Universidade de Coimbra [1] por ter disponibilizado os sensores usados neste projecto assim como referir que a compra do tablet Android foi feita através de fundos provenientes da *European Union's Seventh Framework Programme (FP7)* de acordo com a subvenção n. 3176711.

Palavras-chave

Comunicações sem fios, sensores, plataformas móveis, stress, navegação veicular, reabilitação.

Resumo

Hoje em dia, as pessoas vivem num mundo repleto de situações de stress. Estas podem ter origem em diversos contextos tais como a entrega trabalho a curto prazo, morte de um familiar ou até mesmo com relacionamentos pessoais. É um facto que o comportamento humano é influenciado pelo nível de stress duma pessoa em termos de racionalidade. A título de exemplo, o stress pode ter impacto no contexto da condução de veículos, i. e., pode colocar em risco a segurança dos condutores e das pessoas envolventes.

Neste domínio, o desenvolvimento de um sistema capaz de detectar stress pode, no futuro, apoiar na condução de modo a providenciar uma maior segurança para todos os intervenientes. Um sistema complexo como este pode também ser útil para fazer um levantamento estatístico sobre pontos negros nas estradas, i. e. locais onde os condutores apresentam sistematicamente valores altos de stress, consubstanciando zonas onde tipicamente ocorrem acidentes.

O objectivo deste trabalho consiste em dar os primeiros passos para o desenvolvimento dum sistema capaz de detectar stress em seres humanos de modo a ser aplicado em contextos veiculares tais como na condução automóvel e em sistemas de navegação de cadeiras de rodas elétricas. Neste último, será direcionado para pessoas com elevado grau de limitações de movimento dos seus membros, por exemplo tetraplégicos. Constituindo o primeiro passo para a criação dum sistema complexo, este trabalho apresenta uma plataforma para aquisição de sinais de stress e registos de ocorrências os quais podem ser observados de forma síncrona. Toda esta informação será guardada para permitir a sua análise posteriormente de forma a desenvolver e validar algoritmos de stress robustos e eficazes.

Keywords

Wireless communications, sensors, mobile platforms, stress, vehicular navigation, rehabilitation.

Abstract

Nowadays, people live in a world with a lot of stressful situations. These can come from different contexts as deadline jobs tasks, death of a family member or even problems with personal relationships. It is also known that the human behaviour is influenced by the stress level of a person in terms of rationality. As an example, this can have impact on vehicular navigation, i. e., it can jeopardize the safety of drivers and all other persons around.

In this domain, the development of a stress detection system for vehicular applications can, in future, assist the driving in order to provide a higher safety. A complex system like this can also be used to report relevant and statistical information about *black spots* on the roads, i. e. locations where the drivers present systematic high levels of stress, e. g. accidents.

The aim of this work is to serve as the base of a stress detection system to be used in vehicular applications such as driving cars or electric wheelchairs. In this last context, it will be given focus on severely impaired persons. Being the first step of such system implementation, this work presents a platform that acquires stress signals and register occurrences. This data can then be accessed off-line to do an extensive analysis in order to develop and validate stress algorithms.

Contents

List of Figures

List of Tables

Symbols and Abbreviations

ADC Analog to Digital Converter vii, 16–18

ADT Android Developer Tools vii, 18

ANS Autonomic Nervous System vii, 4, 52, 58

bpm beats per minute vii, 33, 38, 51

csv comma-separated values vii, 28

ECG Electrocardiogram vii, 6, 11, 13, 15, 17, 26, 27, 32–34, 37–41, 43, 52, 55, 58

GSR Galvanic Skin Response vii, 5, 6, 11, 13, 17, 27, 28, 32, 34, 37–39, 41, 50, 52, 53, 55

HR Heart Rate vii, 6, 21, 27, 32, 33, 37–40, 42, 43, 46–49, 51–53, 55, 58

HRV Heart Rate Variability vii, 6, 36, 46, 48, 49, 52

PNS Parasympathetic Nervous System vii, 3, 6, 48, 49, 52

SC Skin Conductivity vii, 5, 15, 17, 21, 27, 32, 37, 39–41, 45–53, 55

SNS Sympathetic Nervous System vii, 3, 6, 46, 52

UI User Interface vii, 24, 27, 32

Chapter 1

Introduction

1.1 Motivation, context and objectives

The sensation of well-being is a special characteristic that any human tries to reach or maintain at any time. However, in some situations, the human being exceeds its homeostasis limits due to several reasons such as less pleasant situations that can happen in job or relationships with other persons. These kind of situations can lead a person to a state of *stress*, and due that, they are called as *stressors*.

Therefore, the development of a stress detection system would be useful to be applied in all contexts that can put in risk the emotional and physical integrity of persons. Considering a car driving as an example, it would be of value that the system detects the level of stress of the driver and, in case it is high, it alerts the driver and advises him/her to reduce the car velocity or suggests a relaxing song for example. Otherwise, this level can increase and the driver can compromise the safety of all persons involved, since it is known that the larger are these levels, the worst will be the rationality of his/her behaviours or actions. Yet on this topic, the usage of such system could help the detection of dangerous points on roads and motorways, i. e., *black spots* - road locations prone to accidents.

This know-how can be also applied in other contexts as in rehabilitation, specifically to aid in the wheelchairs navigation. The tetraplegic persons which are severely impaired present a lot of difficulties when trying to achieve tasks as crossing a door. As long as they do not succeed it, they will get more stressed and there will be a time in which they will give up. Thus, to improve the person's welfare, the stress detection system could be included in the powered wheelchair system in order to provide a semi-automatic navigation that aids the person achieving the intended task. In this case, and besides the stress detection component, the powered wheelchair will also include sensors which acquire environmental data to perform the correct movements.

Relating with the contexts above, it has been developed in the Technical University of Wien a neuro-psychoanalytic model ARS (Artificial Recognition System) which is based on the structural Freud model and can be simplified by the figure 1.1.

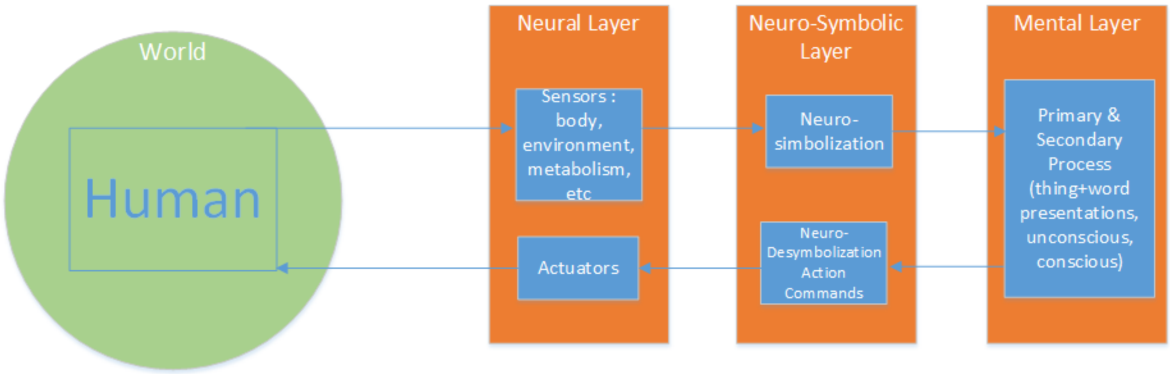


Figure 1.1: Simplified neuro-psychoanalytic model ARS developed in Vienna University of Technology - based in [2].

The final purpose of this project is to develop a stress detection system to include in the ARS model in order to aid severely impaired persons in driving electric wheelchairs and also to be used in vehicular applications, in particular to detect *black spots* on roads before accidents happen. However, the development of a full-blown system with this complexity has to be split in several parts: understand how the human stress can be measured, collect an increased amount of real experimental data, develop and validate stress detection algorithms and then assemble all the relevant components. This dissertation is related with the first two parts, i. e., it consists in the development of a platform that acquires human stress signals and allows the registering of occurrences. By doing a temporal association between the stress signals and occurrences, it will be possible to identify which occurrences can cause more stress. The platform is intended to also store all experimental data into a database which can be accessed in a posterior phase of this work to be analysed off-line. In this off-line processing, it will be possible to study and classify the level of stress according to the circumstances that the subject was dealing and therefore verify the adequacy of the stress detection system.

1.2 Document structure

This dissertation is split in five chapters:

- In the first chapter is given an introduction about this project as well as the motivation to do it and also in which context it is inserted. The main objectives of this work are also referred.

- The second chapter exposes general concepts of stress, how it can be measured in humans, which sensors can perform this measurement and also how they work (some of them). To conclude is given an overview of sensors available in the market.

- The third chapter explains in detail all the work done and is split in four main topics. It starts by exposing the requirements of the intended platform and then the necessary elements to build the final solution are chosen. After that, those elements are described. The next topic presents the application organization and details about its development. The last topic exposes a preliminary stress analysing unit which works off-line.

- In the fourth chapter is done the validation of the platform by showing relevant results. Furthermore, experimental results done in vehicular navigation, video game and tilt table contexts are presented. To conclude, some remarks are given about the measured stress signals.

- The last chapter closes this dissertation with the conclusions of all the work and also gives indications for future work for the ongoing of this project.

Chapter 2

State of the art

2.1 Summary

This chapter presents global human stress concepts and also how it can be detected. Then it is given an overview of some sensors able to measure stress and, to conclude, some sensors available in the market to perform it are presented.

2.2 Stress concepts

The concept of *stress* can be interpreted by different points of view. For example from a physicist perspective, it can be seen as the force applied in a certain body [3]. Although this is a simple definition, this project is more related with physio-psychological and emotional human issues that affects the mental state. Generally, *stress* is associated to a negative feeling as discomfort, anxiety, depression, headaches, tiredness or even panic attacks that can be prejudicial to human's health if it is daily present on them. It can be originated from different causes as troubles in work, illness or death of a family member, school examinations, tasks deadlines and so on.

A possible more complex definition of stress consists in a process which involves a *stimuli*, usually denominated by *stressor*, that triggers a psycho-physical response - [3]. This consequent body response, sometimes referred as *fight or flight* response [4], also makes part of the process and basically it is translated by the way how the organism's homeostasis tries to adapt against an unpredictable event with certain characteristics. The aim of this response is to help the body achieving (or maintaining) an internal equilibrate state leading to a well-being sensation. However, as Lazarus and Folkman (1984) said, this is a state that human may be unable to meet [3]. For example in vehicular circumstances, if the driver is not able to solve the situation provoked by a *stressor*, the level of stress will continuously increase. Hans Selye (1976) called this body's reaction against the stress as *General Adaptation Syndrome* [5].

According to [6], the body's response is strongly interrelated with the Sympathetic Nervous System (SNS) activity as it "*is due to a disruption of the autonomic balance because of the body's physical and mental activity (body accelerator); a mobilization of metabolic resources and increased energy expenditure are given during times of stress, arousal or other external challenges, supported by elevated blood pressure and redirected blood from the intestinal reservoir toward skeletal muscles*". Furthermore, the heart rate and the contractile force of heart muscles increase. In the other hand, the Parasympathetic Nervous System (PNS) is related with the "body brake", i. e., rest (relaxed) or digestion times when the energy of body is restored and conserved, leading to a heart rate and blood pressure decrease. The sympathetic and parasympathetic branches constitute the Autonomic Nervous System (ANS).

2.3 Stress measurement

The measurement of stress can be done by one of these three ways: self-reports, behavioural and cognitive functions measures, and medical/biological measures [5]. This project focus on the third one. According to [7], determining the stress by measuring physiological signals is reliable and challenging due to the reasonable evidence from physiological signal. From now on, these signals will be called sometimes as *biosignals* for simplicity because according to [6] "*a biosignal can be defined as a description of a physiological phenomenon, irrespective of the nature of this description*".

In order to detect and evaluate human emotions related to stress, it was used in the past the electroencephalography, image processing to recognize facial expression changes as well as the tone of speech [8]. However, the advances in electronic sensors and medicine areas have respectively enabled the access and interpretation of some physiological signals data in a easier way.

According to literature, the physiological signals mentioned in table 2.1 carry relevant information about the human stress level [3] [7] [9]. Furthermore, and based in [9], the table 2.2 presents a review of some algorithms used for stress detection.

Galvanic Skin Response (GSR)
Electrocardiogram - Heart Rate (HR), Heart Rate Variability
Blood Pressure (BP), Blood Volume Pulse (BVP)
Electromyogram (EMG)- muscle tension
Pupil Dilation (PD)
Skin Temperature (ST)
Respiration Rate (RR)
Electroencephalogram (EEG)- electrical brain activity
Hormonal changes

Table 2.1: Biosignals that can be used for stress detection.

Support Vector Machines (SVM)
K-nearest neighbor (K-NN)
Fisher Linear Discriminant
Gaussian Mixture Model (GMM)
Analysis of Variance (ANOVA)
Bayes Classifier
Fuzzy logic

Table 2.2: Some stress detection algorithms.

The reading at the same time of all or many biosignals has the advantage of the data complementarity which can lead to more accurate results in the stress detection. But, on the other side, the use of many sensors will restrict the subject's movement and even stress him/her due to the amount of sensors used and the sensation of being observed. As this type of stress is an undesired measure for the purpose of this project, just some of them will be chosen with the expectation that the stress level measure will be reliable. A good solution is to find sensors that can give information of more than one physiological parameter at the same time as the photoplethysmogram sensor [6].

Table 2.3 indicates which biosignals were already used in several algorithms to detect and classify stress levels. In this table, the Bayes classifier is not present because it was proved that SVM is more efficient than Bayes - [10]. The GMM classifier was used in [11] for detecting stress levels in a speech signal. It should be noticed that, in some works, the referred algorithms can be cascaded to improve the results [12].

-	**SVM**	**K-NN**	**Fisher**	**ANOVA**	**Fuzzy Logic**
GSR	X	X	X	X	X
BVP	X				
PD	X				
ST	X				
RR		X	X	X	
ECG		X	X	X	
HR					X
EMG		X	X	X	
Reference	[10]	[12]	[13], [12]	[13], [12]	[9]

Table 2.3: Biosignals used in some algorithms.

It is important to remark that the GSR signal is almost always used in stress detection algorithms. According to [9], the GSR and the HR signals used with a fuzzy logic algorithm can detect stress properly with 99.5% of rate.

2.4 Stress sensors

In order to measure the previously mentioned biosignals, specialized sensors are required. In a general view, sensors for human body usage can be classified as invasive or non-invasive. Since the first ones can cause body damage or pain and then put the subject in a stress state, the focus in non-invasive sensors will be given due to the purpose of this project. A brief description of several sensors that measure the relevant biosignals is presented bellow.

Galvanic Skin Response (GSR)

As mentioned before, this is the most used sensor for stress detection applications. A popular instrument which uses this sensor is the lie detector. Also known as electro-dermal response [4] or psycho-galvanic reflex [14], the GSR signalizes the Skin Conductivity (SC) during a stress event. Its working principle consists in the usage of two electrodes placed in different skin regions - usually on fingers - that measure the skin's resistance variation which can distinguish stress levels. This is due to the fact that, when SNS is activated - mental and physical activity -, the skin starts sweating and, as a consequence, turns it as a good (electrical) conductor resulting in a lower resistance value. The inverse situation is valid when PNS is activated, but it must be taken in mind the recovering time of the biological body resources after a stress event.

Figure 2.1 presents an example of a GSR signal and it is clear that the induced stress stimulus provoked a rise in amplitude. In this case, the *stressor* was related with intensive physical exercise.

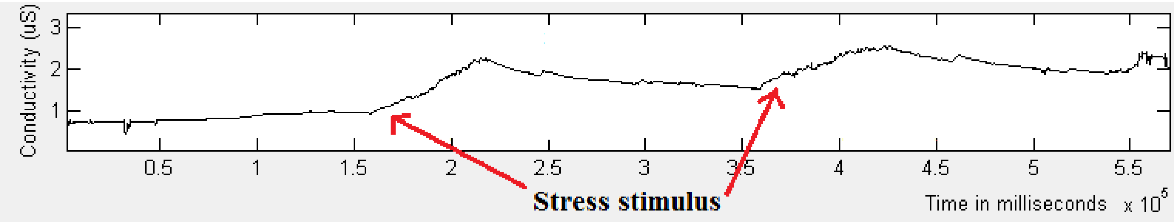


Figure 2.1: Example of a GSR signal.

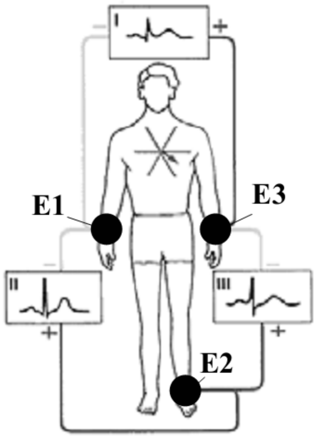
Electrocardiogram (ECG)

According to [6], this biosignal can be classified in three forms: in terms of existence it is considered a permanent signal since it can be accessed at any time without inducing an external excitation; it is also considered a dynamic biosignal due its considerable changes over the time; and finally, concerning its origin, it is an electric biosignal.

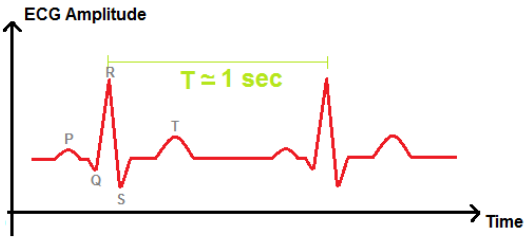
This sensor consists in the installation of electrodes in specific skin places in the body. It is able to measure the electrical excitation of the heart muscles (depolarization) according to the potential difference between the electrodes. In fact, this is the so called cardiac telemetry.

There exists some methods to acquire the ECG signal as the Einthoven-Derivation (frontal-plane) or the Wilson-Derivation (horizontal-plane). Since the last one uses more electrodes, it will be given preference to the Einthoven-Derivation method which is illustrated in figure 2.2a in combination with the typical measured signal in 2.2b. Parameters as Heart Rate (HR) and Heart Rate Variability (HRV) can be easily derived from this biosignal by software. Even the respiration rate can be estimated from this biosignal because the envelope of the ECG is modulated with respiration [6].

The HR is the inverse of the heart period which is given by the time interval between one **R** and the next **R** peak. Concerning the HRV, it describes the fluctuation of the heart period and it is strongly related with the regulatory mechanisms of the SNS and the PNS. In resume, the SNS branch is interrelated with the low frequency (LF) band $(0.04 - 0.15Hz)$ and the PNS is interrelated with the high frequency (HF) band $(0.15 - 0.4Hz)$ of the spectral function of the heart periods [6].



(a) Source: Eugenijus Kaniusas - TU Wien lecture notes



(b) Theoretical ECG signal.

Figure 2.2: Working principle of an ECG sensor.

Overview of other sensors

For the compliance of other biosignals as the pupil dilation, muscle tension, brain electrical activity and blood pressure for example, it is necessary to resort to other methods. To observe the pupil dilation, it is needed to record video and then apply image processing algorithms. To observe the EEG, a cap with several electrodes must be applied in the head. This last type of sensor is discarded because it would have a bad influence on the subjects of this project. The EMG is a good signal to be used in navigation control scenarios as wheelchairs because it reflects the force applied for example in a joystick. This sensor consists too in the appliance of electrodes and then measuring the across voltage which represents the electrical activity of the muscles that cause voluntary body movements. For the blood pressure, it is usually used the photoplethysmogram (PPG) sensor which is installed in the patient's finger. This sensor has the advantage that it can relate several parameters at the same time as heart beat - and by consequence HRV too -, respiration rate, blood pressure and blood oxygenation (saturation).

Available sensors in the market

The following table presents a list of sensors that were found in the market together with their respective price, when available.

GSR	
Companie - Product	**Price in €**
[15] Shimmer - Wireless GSR Sensor	147
[15] Shimmer - Wireless GSR Sensor + Bluetooth	346
[16] Affectiva - Q Sensor (GSR, ST, Bluetooth, Movement Measure)	Unknown
[17] Bio-Medical Instruments - SA9309M	227
[17] Bio-Medical Instruments - Remote Finger Sensor for GSR	28
[18] IndiaMart - Sc/gsr Sensor	Unknown
[19] BioPac - TSD203 & GSR100C	Unknown

ECG	
Companie - Product	**Price in €**
[15] Shimmer - Wireless ECG Sensor	147
[15] Shimmer - Wireless ECG Sensor + Bluetooth	346
[20] Zephyr - BioHarness™3 + Bluetooth	363
[20] Zephyr - HxM™Smart + Bluetooth	62
[21] Polar - Polar WearLink®+ Bluetooth (Heart Rate)	69.90
[22] Quirumed - 127-109605 + Bluetooth	2420
[22] Quirumed - 604-3100BT + Bluetooth	1895
[17] Bio-Medical Instruments - T9306M	227
[18] IndiaMart - Channel Cables For ECG	Unknown
[19] BioPac - TSD155C	Unknown

BVP (Photoplethysmogram - PPG)	
Companie - Product	**Price in €**
[22] Quirumed - 550-SABPM50P	82.50
[17] Bio-Medical Instruments - SA9308M	227
[18] IndiaMart - Blood Volume Pulse Sensor	Unknown
[23] EC21 - Heart Rate Monitor, finger/wrist + Bluetooth	Unknown
[19] BioPac - TSD200 & PPG100C	Unknown

EMG	
Companie - Product	**Price in €**
[15] Shimmer - Wireless EMG Sensor	147
[15] Shimmer - Wireless EMG Sensor + Bluetooth	346

Skin Temperature	
Companie - Product	**Price in €**
[16] Affectiva - Q Sensor (GSR, ST, Movement Measure, Bluetooth)	Unknown
[17] Bio-Medical Instruments - SA9310M	161
[18] IndiaMart - High Resolution Skin Temperature	Unknown
[19] BioPac - BioNomadix Skin Temperature	Unknown

Respiration - Oxygen Saturation	
Companie - Product	**Price in €**
[20] Zephyr - BioHarness^{TM}3 + Bluetooth	363
[17] Bio-Medical Instruments - SA9311M	227
[18] IndiaMart - Oximetry Sensor	Unknown
[18] IndiaMart - Respiration Sensor	Unknown
[19] BioPac - BioNomadix Respiration Transducer	Unknown

Chapter 3

Requirements and development of the data acquisition platform

3.1 Summary

In the previous chapter, the conceptualization of the human stress was done as well as the way that it can be measured. In the following of this line, this chapter aims to describe all the work done in this dissertation in order to accomplish a final platform that acquires stress signals and register occurrences. It starts by explaining the requirements of such a platform, then a description of which components were used and finally how it was built in order to be used by any person. Additionally, it is given an overview of an user-interface that was mainly done to validate the platform and analyse experimental results.

3.2 Requirements

As mentioned previously in chapter 1, the development of a system capable of measuring and classify the stress level of a person can be helpful to improve the safety and well-being of the subject in several contexts.

This work aims to develop a platform able to collect and store stress related signals, as well as register stress-inducer events, on vehicular applications (e. g. electric powered wheelchairs and cars). The captured data will be stored into a local database, which can then be accessed in order to carry out an off-line analysis. This work is intended to serve as the basis for a full-blown stress detection system. So, an increased amount of data has to be collected in real driving situations to perform a complete analysis and therefore validate stress detection algorithms.

As explained in 2, the level of stress can be evaluated through the analysis of several physiological signals, such as ECG, GSR and blood pressure. For the acquisition of these signals, specific medical sensors are required. However, the vehicular environment imposes constraints in the choice of these sensors. These constraints are described bellow:

- Portable
- Non-invasive
- Low price

Whilst driving, the drivers need a high level of freedom in their arms movements in order to control de car. If this freedom is compromised due the amount of cable used by sensors, the driver's security is in risk. Hence, for each sensor, there is a need of portability characteristics such as easy wearable, small size and weight, lowest possible amount of wires used and also a large enough battery lifetime which is important for long time experiments. Therefore, there is a natural inclination for the use of miniaturized wireless sensors. On the other hand, the sensors must be non-invasive to not interfere in the stress measurements since it can cause body pain and then raise stress level of the person by means of physiological signals changes. The price of sensors has also to be took in mind due the budget available for this project at the moment.

Regarding these requirements together with the available sensors in the market (enunciated in section 2.4), only those with wireless characteristics can be evaluated. Although some of them are capable to collect and store data, there is lack of a processing unit to make the temporal association between the physiological signals and the stress-inducer events. The purpose of register occurrences is to understand what kind of events can provoke higher stress levels and then threaten the driver's safety. This is not possible to analyse if there exists no synchronization between biosignals and stress occurrences.

The addition of a processing unit implies a wireless communication interface between the sensors and the processing unit itself. Hence, the initial requirements for the processing unit are described bellow:

- Able to receive data from several sensors via a wireless communication protocol.
- Allow the registering of occurrences in a friendly way
- Capability of data storage

Up to now, the platform overview can be illustrated by figure 3.1.

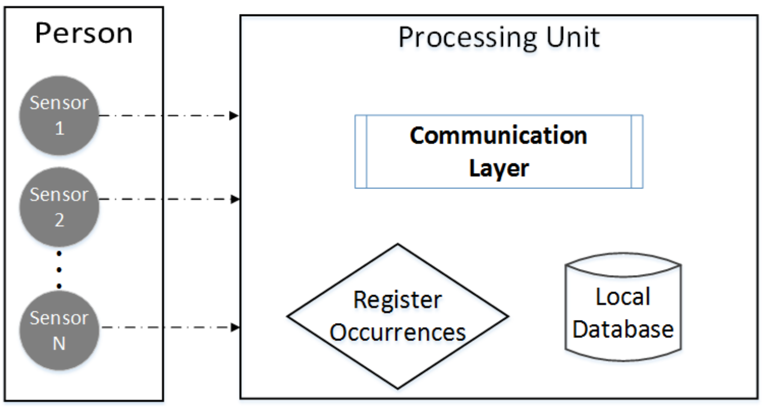


Figure 3.1: Initial block diagram of the platform.

Coming back to the sensors found in the market, some of them had to be chosen in order to acquire reliable stress signals. From those elements it is clear that many sensors are offered with a very high price which is not within the project budget. Some of them, simply don't support a wireless communication interface. Other suppliers don't show the prices on the internet. The request for a quote would jeopardize the timeliness of this project.

Regarding the others, and since the comfort of the wearable sensors is crucial for this project, it was decided to work with the ECG and GSR sensors from the *Shimmer-Research* company[1] which were specifically developed for research purposes. This is a good starting point not only because they already exist in our laboratory but also because they can give some of the most relevant information about the human stress level according to the previous chapter study. In the future work, a more accurate analysis will be done in order to select the sensors for a more definitive platform.

The chosen sensors use Bluetooth technology as the communication interface. Therefore, it is mandatory that the processing unit also supports this technology in order to receive data from sensors in a wireless way. Combining this last requirement with those previously mentioned for the processing unit, it was decided to use an Android device. With this component, it is possible to create an user-interface application that satisfies all the requirements for the processing unit. Beyond that, all the software tools necessary to develop an Android application are open source which implies a lower cost and enables a quick start development.

Considering all the elements chosen, an overview of the platform to be developed is illustrated in 3.2. Hence, the specifications of the user-interface application can be clearly defined. This application is intended to be used by any user, so it is recommended that it presents characteristics such as user-friendliness and robustness (e. g. eventual bad user input).

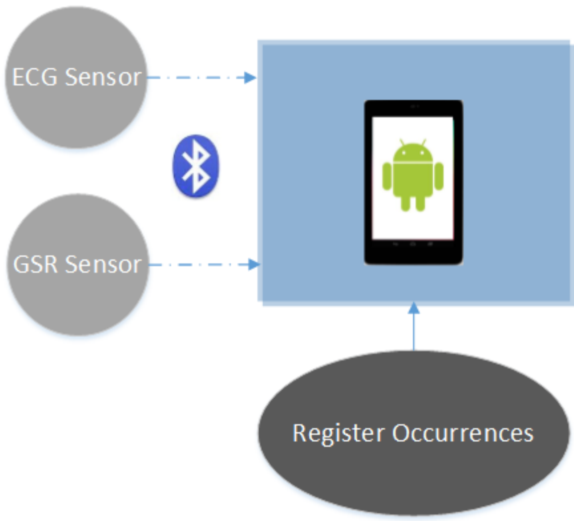


Figure 3.2: Platform overview - system to be developed.

Apart from establish a wireless communication with sensors, the registering of occurrences has some requirements. It has to be possible to register the timings of two types of occurrences: **normal activities** and **spontaneous events**. The idea of the normal activities arose due the fact that it would be interesting to analyse the subject stress signals in specific road map locations and then try to evaluate which zones can be considered as *black spots*. For example, these locations can be cities, counties or motorways. Hence, it was decided that each normal activity has to be registered with a begin and an end moment without overlapping with other activities. Concerning the spontaneous events, they can be registered at any time and they are intended to mark situations such as traffic lights changes, pedestrians crossing road suddenly,

[1]Shimmer-Research is a division of Realtime Technologies Ltd [15].

passings or other unexpected events. Figure 3.3 illustrates a hypothetical experiment in which these occurrences are marked and consequently being associated with the stress signals that can vary or not. The occurrences registering has to guarantee a temporal resolution of at least one tenth of a second. This is, a priori, a satisfied requirement since the Android device permits to access temporal instances in the order of milliseconds.

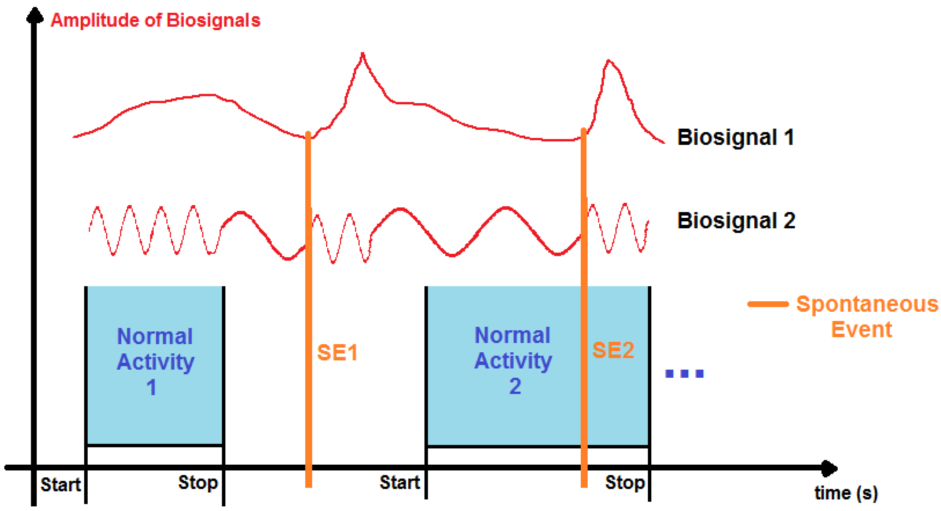


Figure 3.3: Hypothetical visualization of biosignals data and occurrences.

Concerning the experimental data storage, it was decided to organize the local database as is illustrated in figure 3.4. The application has to give the possibility to register each experiment with the patient name. Each folder will contain files that carry all acquired data of the respective experiment.

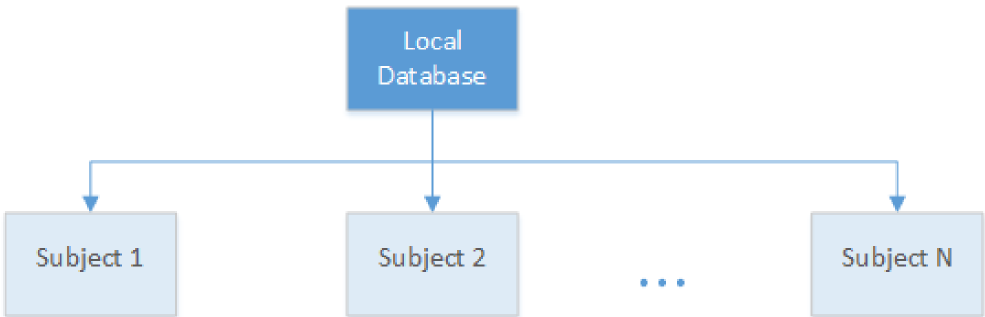


Figure 3.4: Local database organization.

Additionally, it was decided to display values captured by the sensors in real-time. Thus, the user receives a visual feedback of the biosignals that varies among the experiment.

Regarding already the future offline analysis, it was created a Matlab [24] interface that exposes the measurements taken and also gives statistical results. The aim of this interface was to help the validation of the required platform and also to be the first prototype of an analysing unit. A full-fledged version of this unit will perform an off-line processing to verify the adequacy of the stress detection system. However, this would require a constantly and manual file transfer between the Android device and a remote computer that holds the Matlab software. To eliminate this inconvenient, it was decided to add a new functionality in the Android application which allows the uploading of data to Dropbox [25]. Although this feature requires a internet connection on the Android device, it easily carries out the file transfer enabling a quick access of the measurements on the analysing unit.

In resume, the requirements for the Android application are described bellow:

- Button to connect and disconnect with the wireless sensors and signalize the current state (connected or not).
- Button to register a new patient.
- Button to initiate and stop an experiment.
- Buttons to register occurrences with a resolution of one tenth of a second:

  1 to Normal activities.

  1 to Spontaneous events.
- Button to upload all data to Dropbox.
- Button to empty the local database.
- Display the captured biosignals in real-time.
- Display general informations in a panel.
- Button to clear the panel of informations.

3.3 Hardware elements

In this section is given an overview of the commercial sensors and also the processing unit used in this project. In the first ones, it will be described their configuration, what values they transmit, the communication protocol and also how they are applied in the human body. About the last one, it will be described its characteristics and also the tools to develop the application.

3.3.1 Sensors

In order to acquire ECG and SC signals individually, it was used the commercial sensors developed by Shimmer-Research - model Shimmer 2R. Favourable to mobility situations, each sensor includes a 3-axis accelerometer[2], a Bluetooth module RN-42 from Roving Networks and a $450mAh$ rechargeable Li-ion battery.

They can be configured with different sampling rates, depending on the signal properties and the desired signal quality. This was done by using the *Schimmer Connect v0.2* desktop software which permits to select sampling rates from $10Hz$ to $1000Hz$. Depending on this rate, each sensor associates a time stamp for each acquired sample and it can assume values from 0 to 65535 (two bytes) which represents the number of ticks coming from the internal sensor clock [26]. This last component ticks 32768 times per second which allows the calculation of

[2]For now, it was decided to not use the 3-axis accelerometer.

the elapsed time between samples. A special attention must be paid to the fact that the time stamp value can overflow.

Other parameters can be configured depending on the sensor. According to [27] and [28], these sensors have a 12-bit Analog to Digital Converter (ADC) which means that the measured biosignals can assume values from 0 to 4095 which can be converted next into its real units.

Concerning now the sensors protocol to communicate with another remote device (*receiver*) after guarantee a previous connection[3], it can be simply represented by the figure 3.5. It is a simple protocol which consists in a handshaking that has to be done before and after the data transmission. Due that, it is allowed to perform multiple data transmissions without disconnect and connect again with the sensors. More specifically, the handshaking has three command bytes: start (*0x07*), stop (*0x20*) and acknowledge (*0xFF*).

Basic tests such as data frame verification were done on the RealTerm terminal [29] to aid the code development for the processing unit. The sensors have an indicator led that is on when sensors are connected and blinks when they are transmitting data.

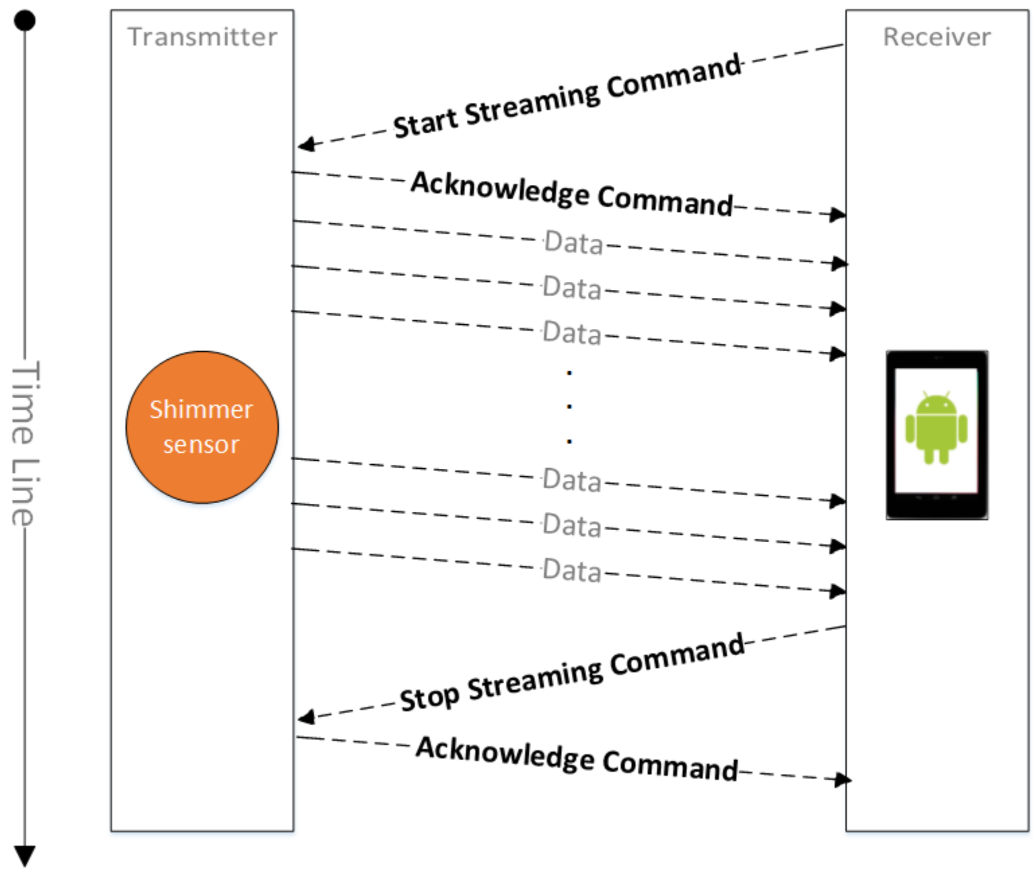


Figure 3.5: Wireless communication protocol between Android device and one Shimmer sensor already connected.

[3]This mandatory connection will be performed by the Android device as explained later.

GSR sensor

As suggested in its manual, the application of the GSR sensor was always done as the figure 3.6 presents. The sensor has two inputs for the electrodes applied in the fingers and it measures the SC between these two points. This measure is the main output and it can vary from $0.2\mu S$ to $100\mu S$. In fact, this range is split in four sub-ranges to perform better measurements and the sensor was configured to be in the *Auto-Range* mode which enables an improved calculation of the current SC level of the subject. Regarding the calculation of the SC, it is done as it follows:

$$SC = a * x + b(Siemens) \tag{3.1}$$

in which x is the value coming from the ADC and the parameters a and b are different constants for each sub-range.

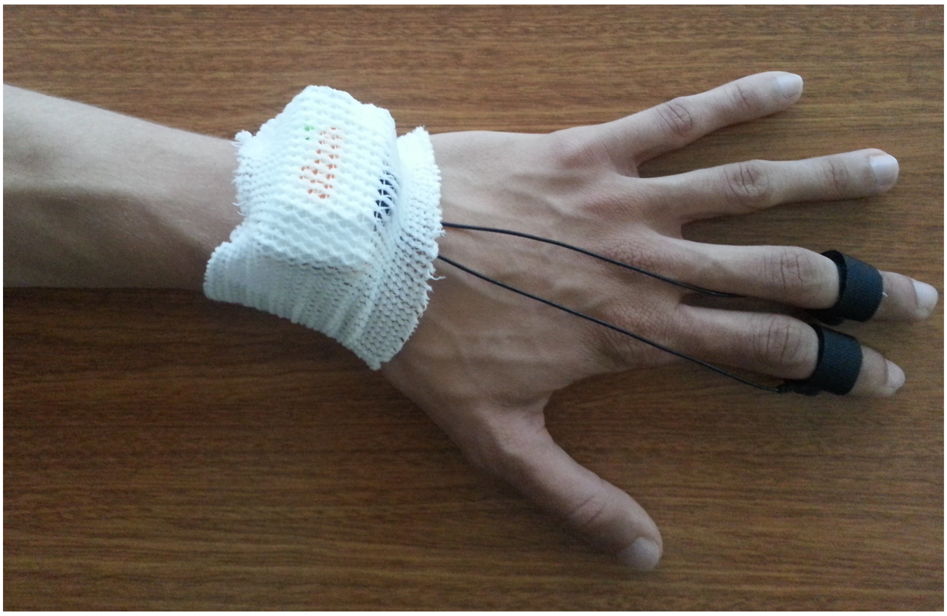


Figure 3.6: GSR sensor application.

Concerning the sampling rate, this sensor was configured for all experiments done with $10Hz$ since the SC does not need to be measured at frequencies exceeding $5Hz$ [4]. Therefore, the time stamp increment from one sample to other should be around 3276 ticks. However it was observed that this difference (in ticks) between consecutive samples was 3200 which means that the sensor samples at $10.24Hz$ in reality.

ECG sensor

As the figure 3.7b illustrates, the application of the ECG sensor in the human body reveals a high level of portability.

(a) ECG sensor - figure obtained from [30]

(b) ECG sensor application.

Figure 3.7: Overview of the ECG sensor.

This sensor has three inputs to apply the respective electrodes: left-arm, right-arm, and left-leg. Its main outputs are the values of lead II (right-arm, left-leg) and lead III (left-arm, left-leg). The value of lead I (right-arm, left-arm) can be calculated by software by subtracting lead III to lead II. The calculation in mV of the each lead is done as it follows:

$$Lead(mV) = \frac{ADC_{value} - ADC_{offset}}{Gain} * ADC_{sensitivity} \tag{3.2}$$

By default, the offset is 2060 and the gain is 175. For more accurate results, these values have to be measured experimentally - [28]. About the sensitivity, it is given by:

$$ADC_{sensitivity} = \frac{V_{ref}}{max\ ADC\ value} = \frac{3000(mV)}{4095} \tag{3.3}$$

Regarding the sampling rate, it was observed, by doing several experiments, that $250Hz$ was enough to give a good signal quality in order to detect the typical PQRST wave - figure 2.2b. Selecting this rate, it means that the time stamp increment from one sample to other should be approximately 131 ticks. However it was verified that the difference (in ticks) between consecutive samples was 128 which means that the sensor samples at $256Hz$ in reality.

3.3.2 Processing unit

The processing unit used was the tablet Asus Nexus 7 [31] which comes with the Android operating system 4.2.2 (Jelly Bean - second higher platform version distributed after Gingerbread according to [32]). It is suitable for this project since it has the required characteristics previously mentioned in section 3.2 and also has the advantage of being supported by Google i. e., the software updates are done more quickly. Besides that, there exists a large community of software developers that augment the development support due the fact it is an open source project, free and multi-platform. As it was recommended in [33], the development of the Android application was done using the Android Developer Tools (ADT) Bundle which comes with an Eclipse IDE[4] [34]. The programming language used is Java.

The tablet comes with a Bluetooth interface (version 2.1) that enables the establishment of a communication with the two wireless sensors and also $32GB$ of storage to save experimental data. Additionally, this processing unit includes the Wi-Fi 802.11 b/g/n technology.

[4]Integrated Development Environment.

Figure 3.8: Processing unit - figure source [35].

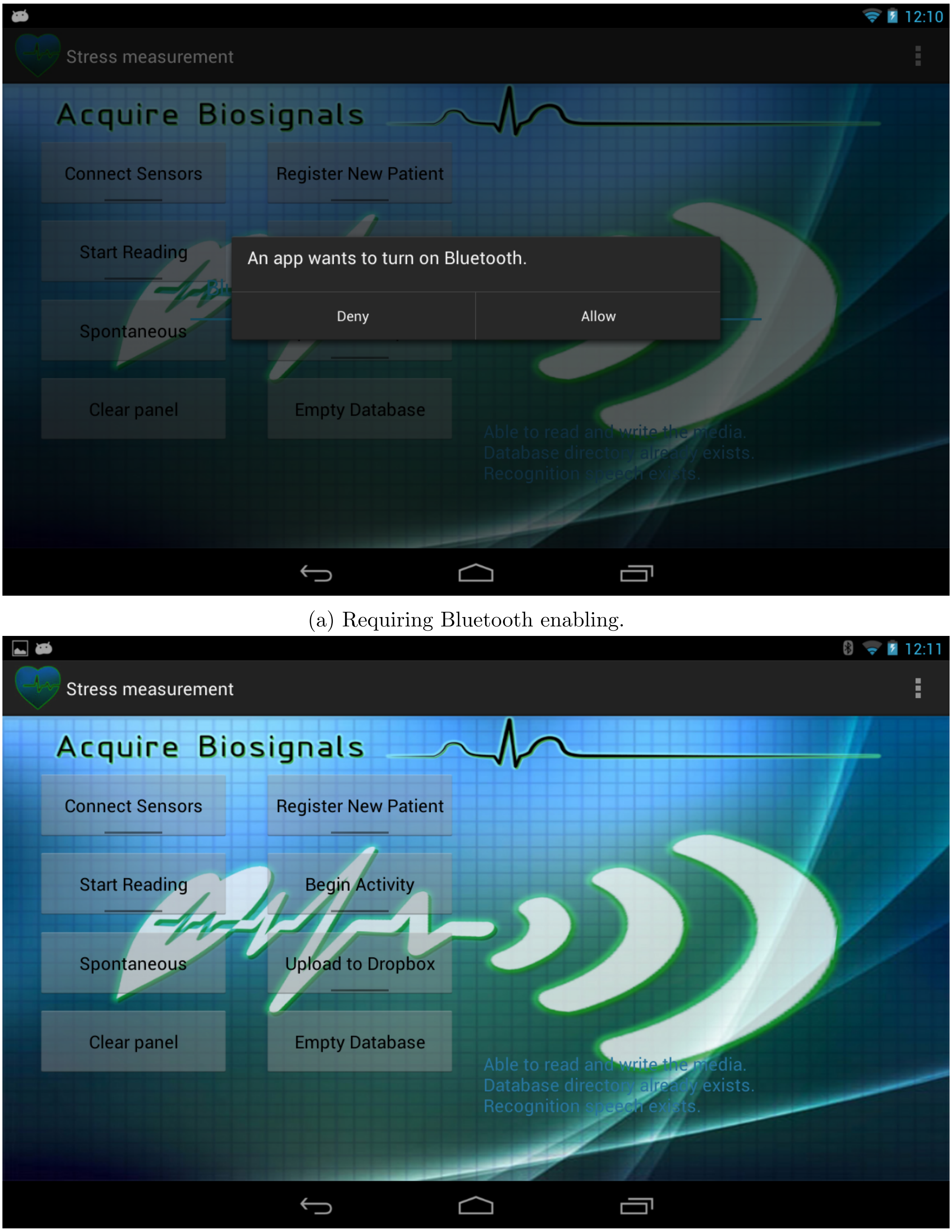
3.4 Application organization and development

The development of the Android application, called *Stress measurement*, was an important part of this project since it has to handle all the previously mentioned requirements, it has to be user-friendly and also it has to take care of possible bad user's inputs or internal device errors. Initially, this section presents a quick overview of how the user can interact with the application and then it will be given more detail about its inherent functionalities.

When the application is launched, the first thing that is attempted (without the acknowledge of the user) is the creation of a directory on the device memory, called *Database*[5], that will store all experimental acquired data. Due that, it is done an initial check to determinate if it is possible to write and read the media at the moment. It also verifies the existence of a Bluetooth module and a speech recognizer. The last one was considered a promising feature regarding the vehicular applications. Since it is available on the tablet, it was decided to adapt it for the registering of occurrences. Regarding again the initial check, if any of those conditions fails, an error message is presented to the user and the application is closed.

In case of a successful check, then, the layouts of figure 3.9 are shown up sequentially. First, it is asked permission to enable Bluetooth and after the user allows it, he/she can start interacting with the layout 3.9b. There, it is visible a panel in the bottom-right zone that gives general informations such as the initial check, and it can be erased by pressing the button *Clear panel*.

[5]If this directory already exists, nothing is done.



(a) Requiring Bluetooth enabling.

(b) Initial view

Figure 3.9: Initial views of the Android application.

Figure 3.10 represents the normal flow of the Android application. As is required in all Bluetooth aplications, the first intent in this case is to establish a connection between the Android device and the two sensors. Hence, after pressing the *Connect Sensors* button, the layout 3.11 will not disappear till the connection is not guaranteed. After that, it is necessary to register a new subject by pressing the button *Register New Patient* and then typing his/her name as illustrated in figure 3.12. This register menu will be re-prompted in case that the name inserted already exists in *Database*.

Therefore, the user can initiate a new experiment, by pressing the button *Start Reading*, and register an unlimited number of occurrences by pressing the *Begin Activity* or *Spontaneous* button. At this moment, it can be said that the application is in *modus operandi* which is illustrated in figure 3.13. There can be visualized the biosignals changes in real-time, more specifically the SC (blue line of the graphic) and the instantaneous HR value. As will be explained in subsection 3.4.2, the registering of occurrences can be done by voice or typing.

When the experiment is finished, the user just has to press the *Stop Reading* button and then, all acquired data is saved in the subject directory - see subsection 3.4.1 for more details. Afterwards, he/she can disconnect the device from sensors and then terminate the application, or instead of that, the user can always start a new experiment by registering a new patient and therefore perform multiple attempts.

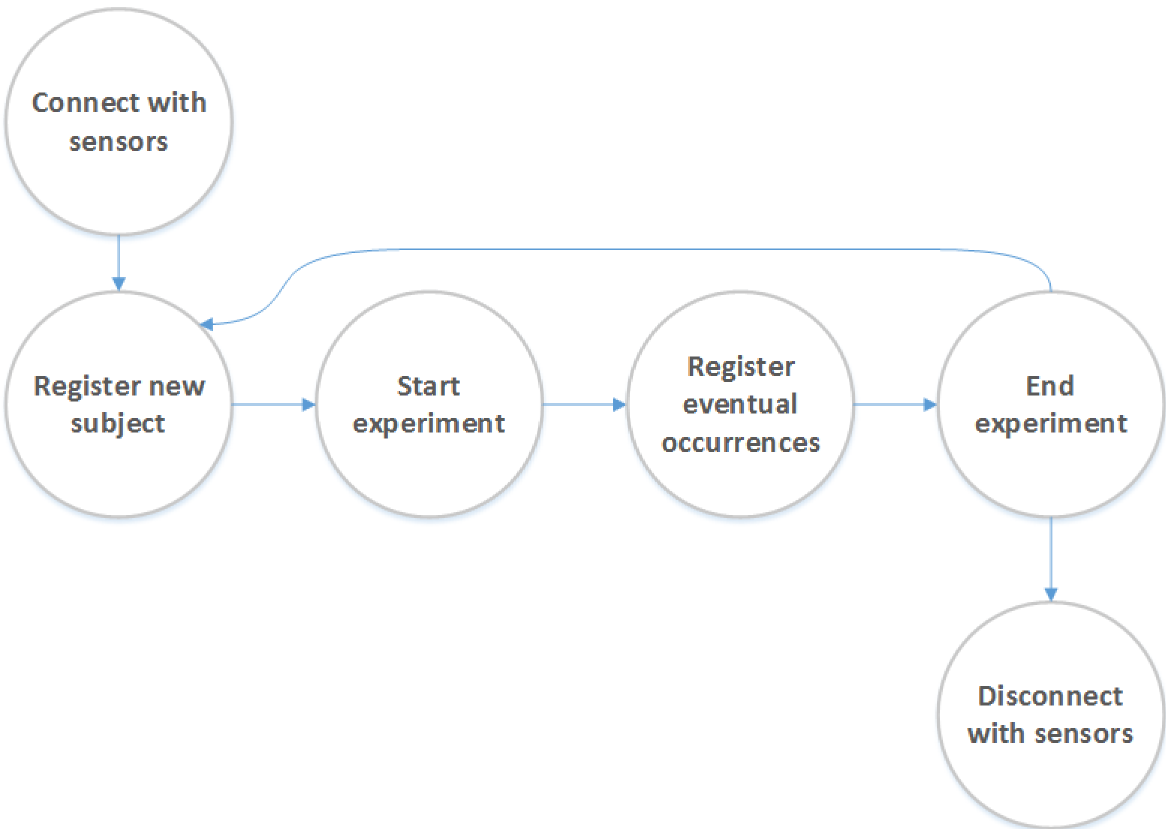


Figure 3.10: Normal flow of the Android application.

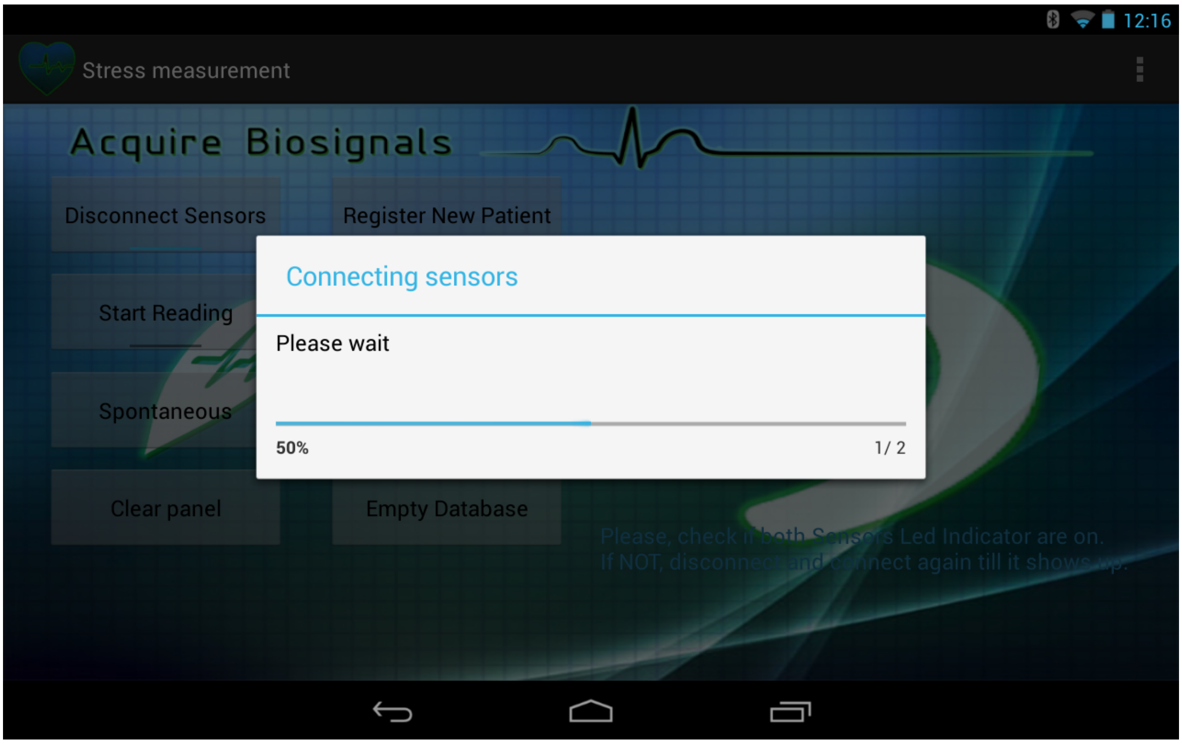


Figure 3.11: Bluetooth connection between the Android and the sensors.

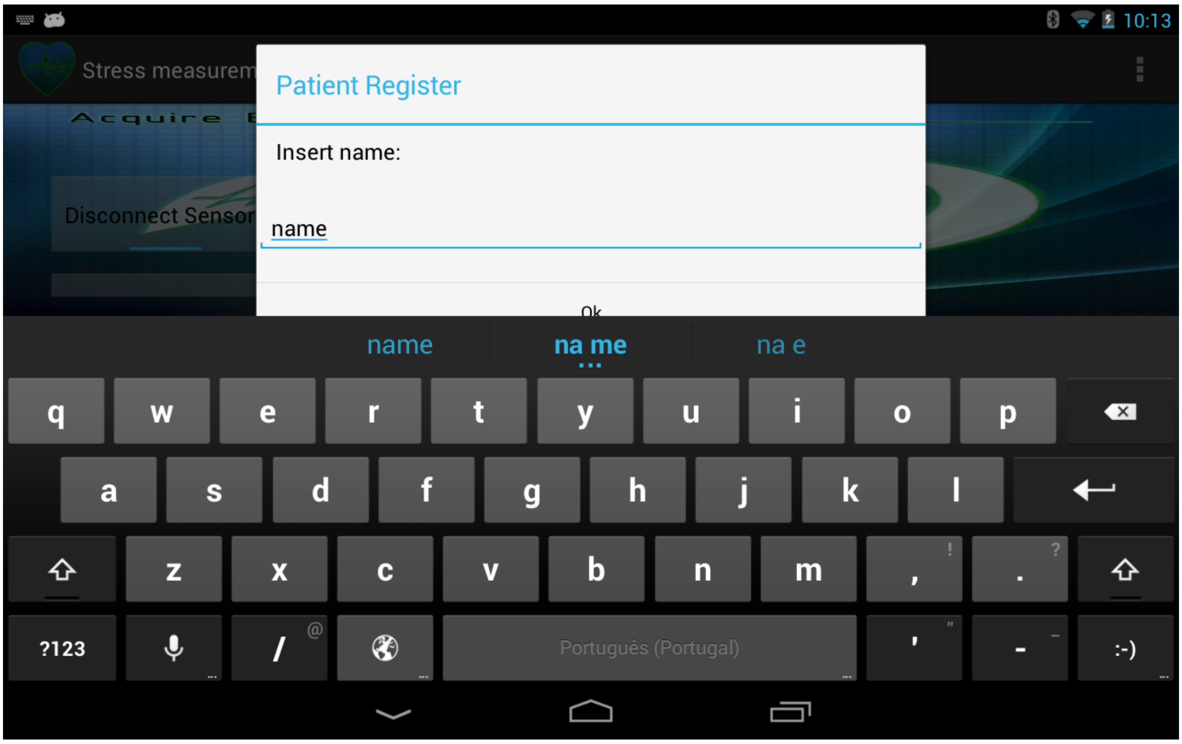


Figure 3.12: Registering a new patient.

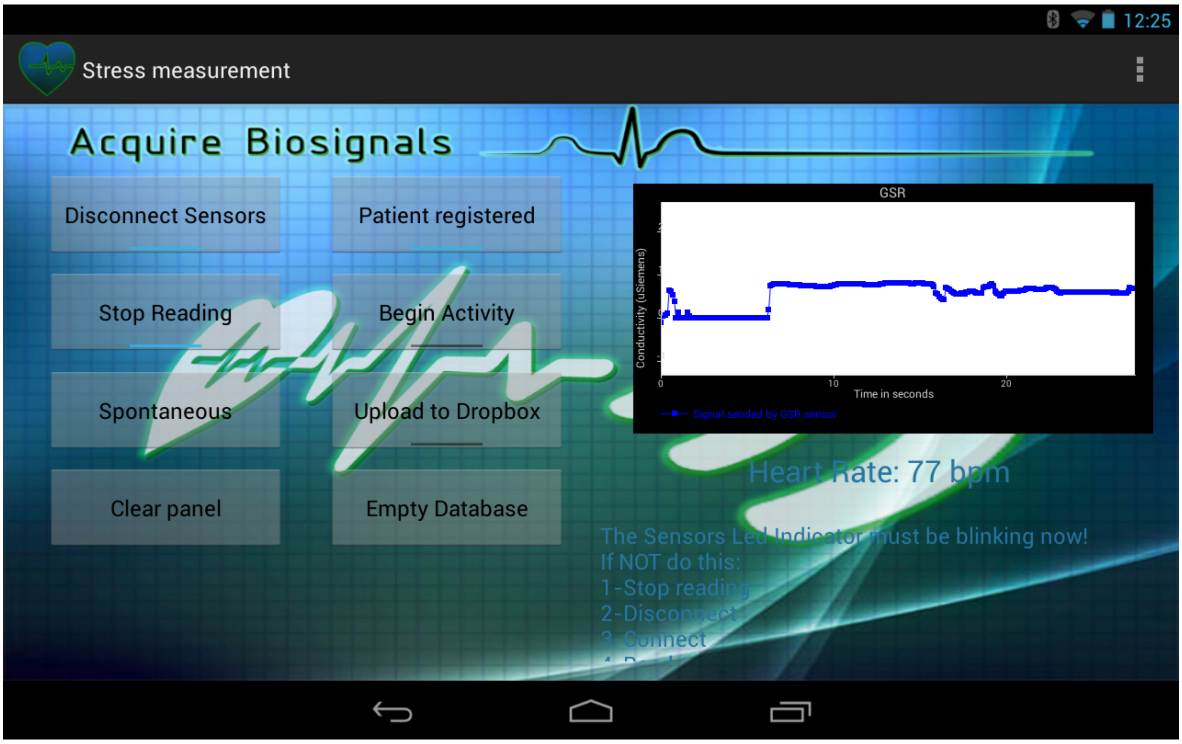


Figure 3.13: Application layout in its *modus operandi*.

Concerning the remaining features, the data uploading to Dropbox is an isolated operation that is explained in section 3.4.3 and the emptying of the *Database* is done after the user confirms it - see figure 3.14.

In the following situations, nothing is performed besides giving an adequate warning message to the user:

- Trying to register a patient without a Bluetooth connection guaranteed.
- Trying to start a new experiment without registering a patient first.
- Trying to register occurrences out of an experiment.
- Trying to end an experiment without ending a current *Normal Activity* first.
- Trying to upload data to Dropbox during an experiment or when *Database* is empty.
- Trying to empty *Database* since after the patient has been registered and the end of the experiment.
- Trying to disconnect sensors during an experiment.

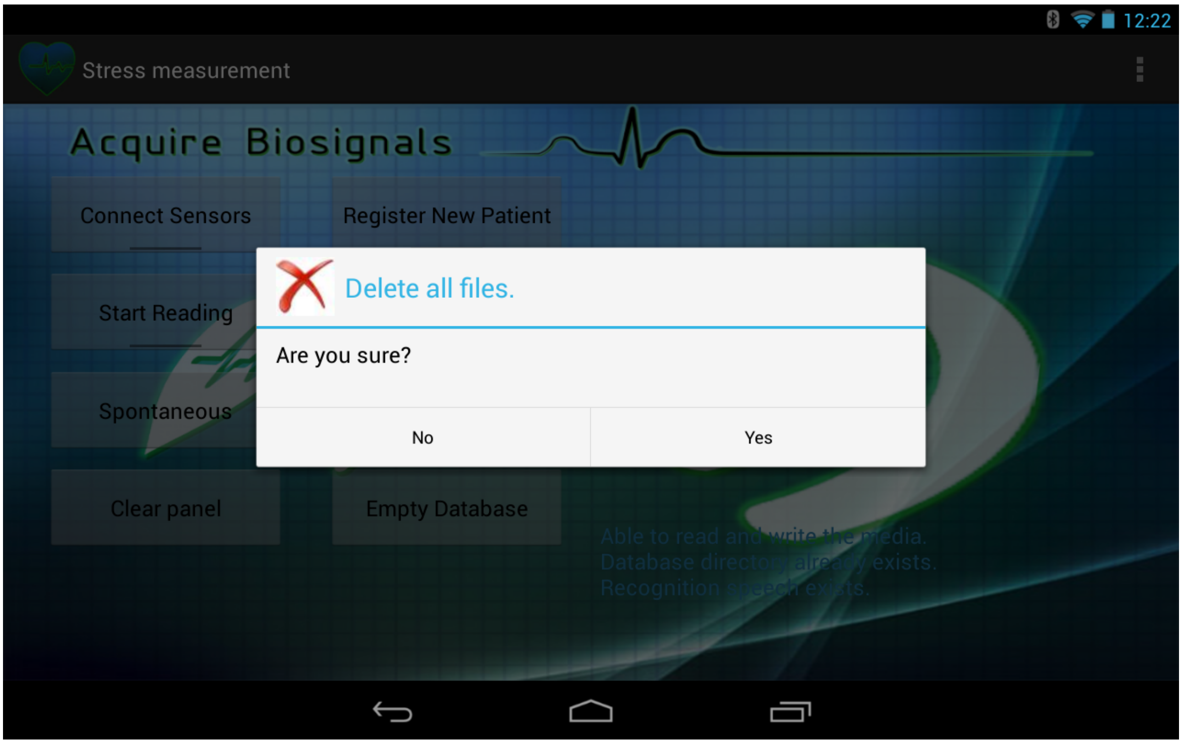


Figure 3.14: Deleting all files from *Database*.

Hereupon, the development of the application will be described in more detail, but before that, it has to be referred that a Bluetooth pairing between tablet and both sensors is required before attempting any connection. This was done manually instead of dynamically[6] because the sensors were known a priori, so the directly usage of their Bluetooth addresses was enough to establish the intended wireless communication.

Concerning the User Interface (UI) elements, this application is essentially composed by widgets (buttons) that allow the user to perform specific operations. Analysing the figure 3.13, the buttons with a line centered in its bottom region are the so called *toggle* buttons, which are suitable to control two complementary actions. This line represents the current state according to the button action. For example, the two top buttons present a cyan line which, in this case, means that the sensors are connected and a patient has already been registered. Another example is that if the user wants to register a new activity, the line turns into a cyan color after he/she presses the "Begin Activity" button. The next time that the user presses this button, this line will return to the black color signalizing the end of the normal activity. In this way, it is guaranteed that an overlapping of this type of occurrences will never happen. The usage of the eight buttons allowed to control the application flow as it is represented in figure 3.15.

[6]Dynamic pairing means searching from Bluetooth devices and choose specific ones to pair.

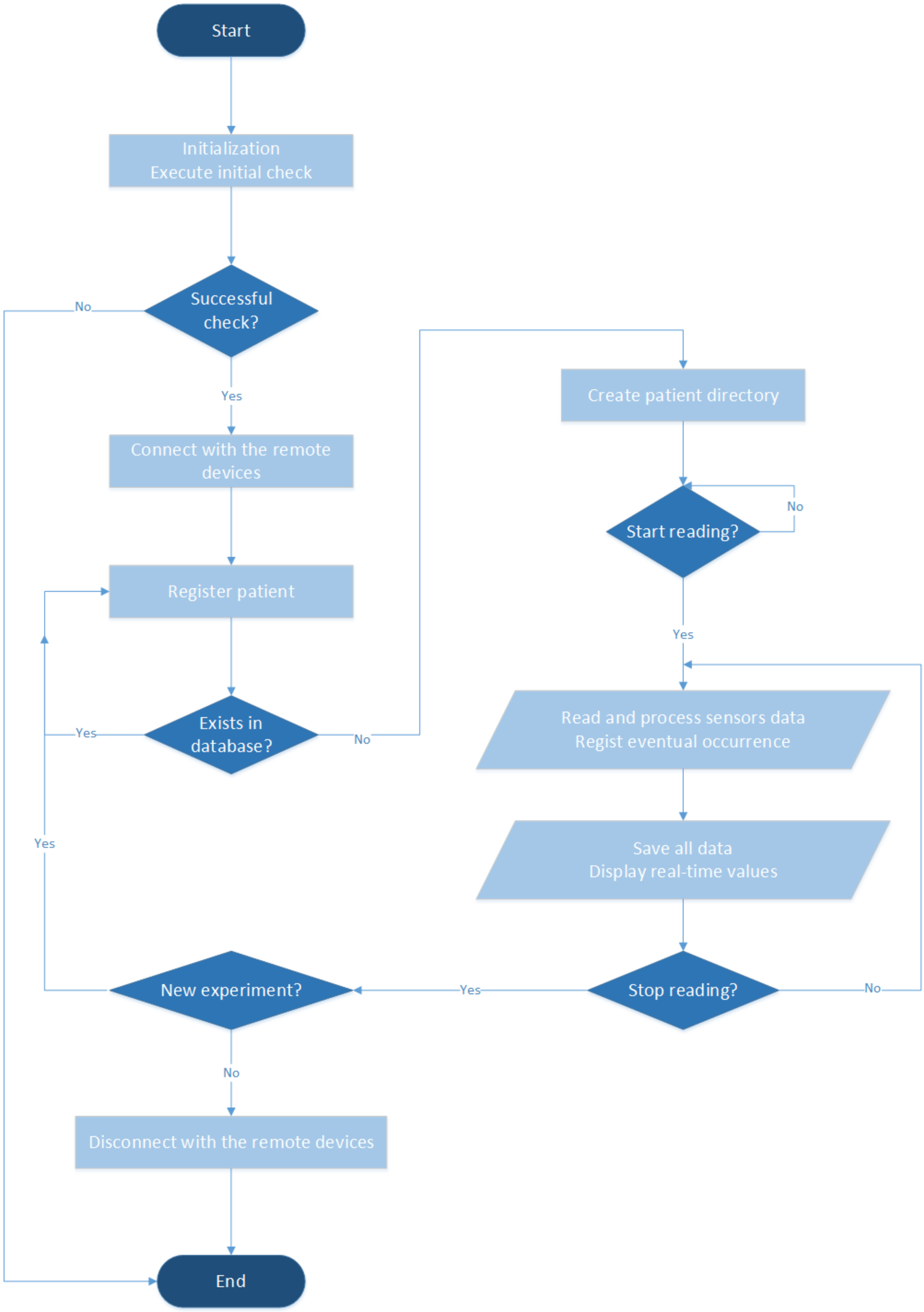


Figure 3.15: Flowchart of normal application's operation.

About the internal structure of the Android application, some concepts have to be introduced first. Normally, an Android application can be seen as a chain of activities[7] that can switch between them, however, only one was used in this application - *Main Activity*. According to [36], an activity is a component that produces a single screen layout enabling users to interact with what is presented on it such as buttons, images, etc. As illustrated in 3.16, each activity has a life-cycle with several states associated to callback methods as *onCreate()*, *onStart()*, *onResume()*[8], *onPause()*, *onStop()* and *onDestroy()*, in order to turn the application more flexible according to specific paths for state transitions. The developed activity only uses the first three methods.

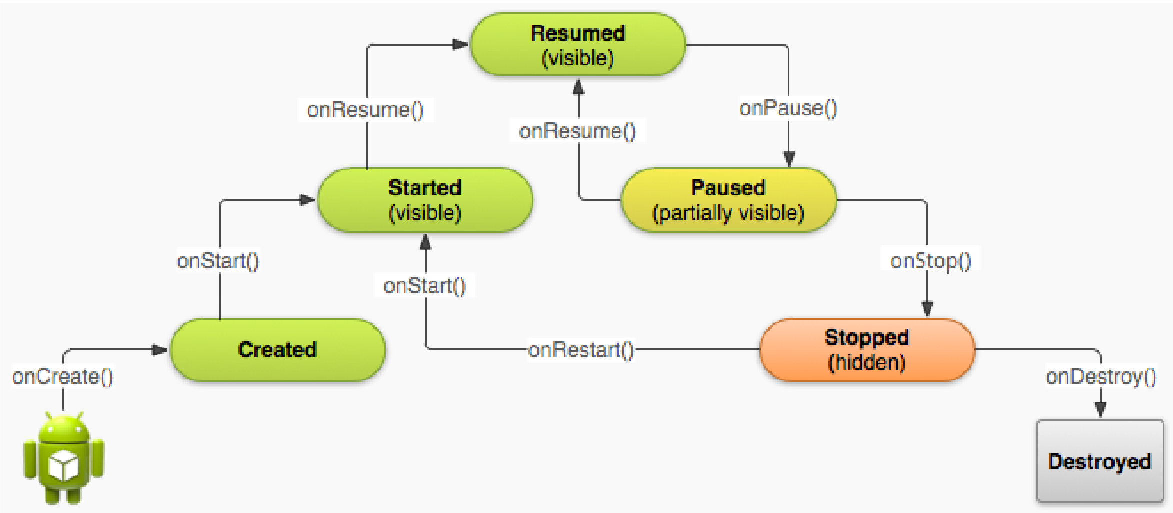


Figure 3.16: Activity life-cycle - figure source [37].

In the *onCreate* method is done the creation of view objects and also there was done the majority of the control component regarding the normal operation of the application - see flowchart 3.15. A welcome message is given to the user when the activity reaches the *Started* state which normally comes after the *Created* and is followed by the *Resumed* state. In the *onResume()* method, Wi-Fi is directly turned on (in case it was not) and for Bluetooth it is asked to the user to enable it or not by launching a new activity using the *startActivityForResult(Intent, int)* method. When it happens, the *Main Activity* switches to the *Paused* state turning its screen layout partially visible as can be noticed in figure 3.9a and, after a positive or negative user input, the *Main Activity* returns to the *Resumed* state again, so it will stay on that loop while the user doesn't allow to enable Bluetooth. When the user grants the permission, the *Main Activity* remains on the *Resumed* state, and then, the user can start interacting with the screen presented at 3.9b as was previously explained.

Concerning the Bluetooth communication, the developed code was based in the Android's Bluetooth APIs and some given examples as the Bluetooth Chat [38]. A particular note for the Bluetooth adapter from the tablet, in which was turned off its discovering resource of remote devices because it is an intensive procedure that is unnecessary in this case. For the connection part, the tablet has to connect (as a client) individually with the two Bluetooth sockets that are created according to the Bluetooth addresses *00:06:66:42:1E:5A* (ECG sensor) and

[7] Do not confuse with *Normal Activity* from the occurrences.

[8] *"Called just before the activity starts interacting with the user."* - [36]

00:06:66:42:1E:7D (GSR sensor). It has to be remarked that these connections are done in an *AsyncTask* [39] that is executed in background and publishes the results on the UI thread[9] as can be seen in figure 3.11. After this, the tablet can handle two independent communication channels which will be managed by *Threads* [41] - one for each.

It was decided to manage each channel using one *Thread* instead of an *AsyncTask* since this last one should be used ideally for brief operations [39]. Both threads, which are executed concurrently with the UI thread, are started when the *Start Reading* button is pressed and then, it is sent the *Start Streaming Command* to sensors. First it is sent for the GSR sensor and then for the ECG sensor. At this stage, a file to save all the occurrences is created and a global variable containing the starting time of the experiment (using the *Date* class) is initialized. Each thread creates a file (*ecg* and *gsr* files) to constantly write on it the relevant data derived from the received Bluetooth frames during the experiment - see subsection 3.4.1. Also during it, the SC and HR are sent to the UI thread in order to visualize their values in real-time - see section 3.4.4 for more details. When the user intends to stop the experiment by pressing the respective button, the *Stop Streaming Command* is sent to each remote device and the files are saved in the subject directory.

Next, will be described the storage data organization, the register of occurrences, the upload of data to Dropbox, the visualization of biosignals in real-time and finally the HR algorithm implemented. But before that, a remark has to be done. Due the huge arrival delay of the tablet, the application had to be tested first in a *smartphone* with the Android version 2.3.3 and only afterwards it was done the code adaptation for the tablet due to the fact that the screen sizes and the operating systems versions are different. Besides the layout, the important change from one version to other was to put the network operation part (upload to Dropbox) out of the UI thread because on newer versions of Android, starting with Android 3.0, it throws the *NetworkOnMainThreadException* [42]. That's the reason why an *AsyncTask* was used.

3.4.1 Data saved in the external storage directory

The data measured in real experiments has to be saved for posterior off-line analysis and in this case it is kept in the Android external storage directory which, according to [43], "*it is a filesystem that can hold a relatively large amount of data and that is shared across all applications*". To cluster there all data coming from this application, the *Database* directory is created and is in this exactly location that the *Empty Database* button also acts. As was said previously, an unrepeated directory is created for each patient before reading from sensors, so, when the user decides to terminate the experiment, all acquired data is stored in the respective folder as the figure 3.17 illustrates.

[9]"*When an application is launched, the system creates a thread of execution for the application, called "main." This thread is very important because it is in charge of dispatching events to the appropriate user interface widgets, including drawing events. It is also the thread in which your application interacts with components from the Android UI toolkit (components from the android.widget and android.view packages). As such, the main thread is also sometimes called the UI thread.*" - [40].

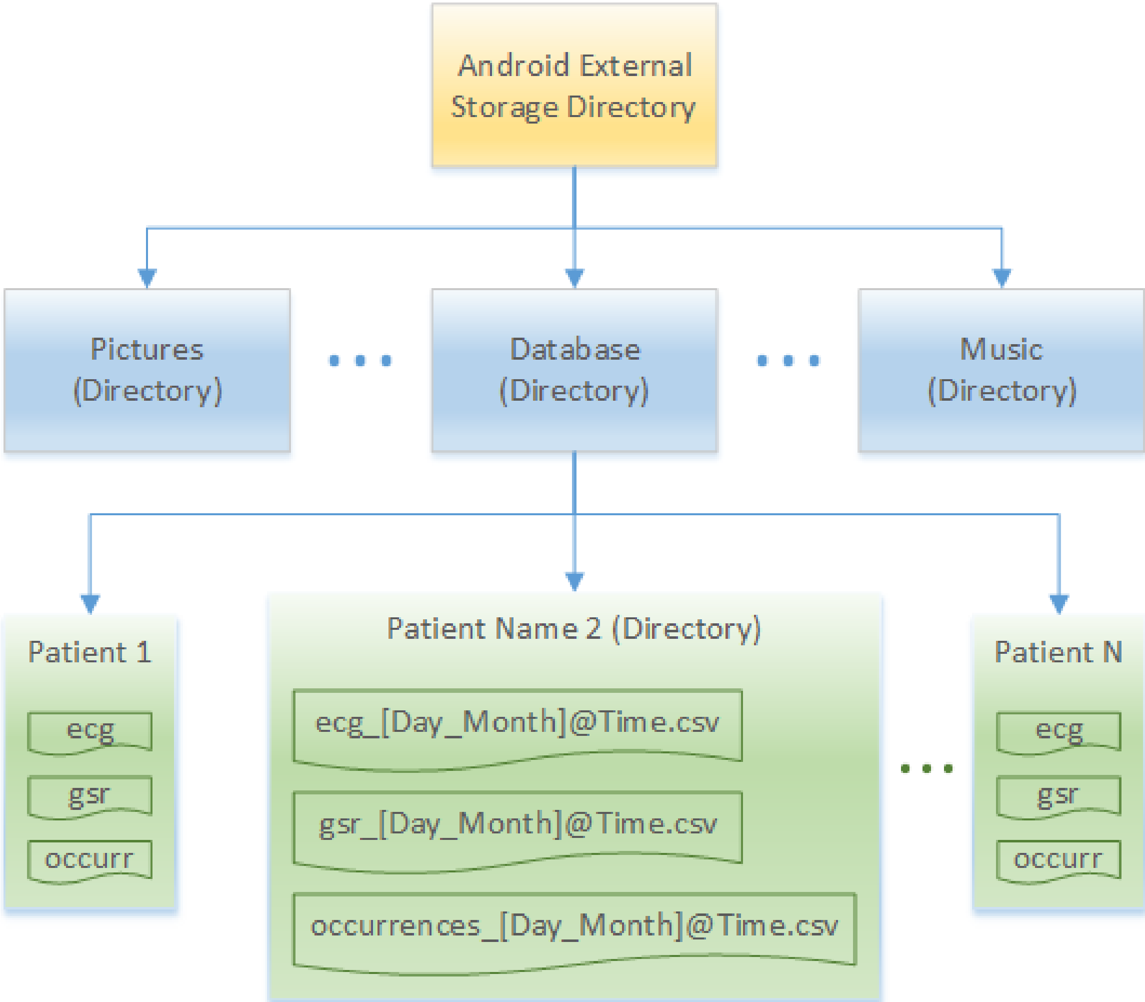


Figure 3.17: Data storage for each patient.

Each patient directory contains three different files: ecg, gsr and occurrences. The name of each file contains the starting date of the experiment: day, month, hour, minutes and seconds - making them unrepeated without taking in mind the year. Each file is saved in the comma-separated values (csv) format which is quite simple and is supported by a lot of software platforms as Matlab[10]. The figure 3.18 indicates, for each file, the headers[11] (first line) which indicates the parameters with their units, and exemplifies some values acquired experimentally. Since the GSR sensor was configured to be in *Auto-Range* mode, in case of the *Range Change* parameter, the value **0** means no change and the value **1** means that a change happened.

[10]To read csv files, Matlab has the function *csvread('filename')*.
[11]In case of the occurrences file, the '*Time (ms)*' field represents the same meaning of elapsed time.

```
Elapsed time (ms), Conductivity (uSiemens), Range Change
0.000000,-0.187000,0
97.656250,0.071600,1
195.312500,-0.018800,1
292.968750,0.001707,1
390.625000,0.002163,0
```

(a) GSR file

```
Time (ms), Occurence name
60119,Spontaneous: a
210057,Begin of: b
420394,Spontaneous: c
765314,Ending of: b
900432,Spontaneous: experiência
```

(b) Occurrences file

```
Elapsed time (ms), Lead II (mV), Lead III (mV), Heart Rate (bpm)
0.000000,-0.355835,-0.200942,0
3.906250,-0.238619,-0.146520,0
7.812500,-0.138148,-0.108844,0
11.718750,-0.117216,-0.087912,0
15.625000,-0.125589,-0.100471,0
```

(c) ECG file

Figure 3.18: Examples of each file saved in a patient directory.

3.4.2 Register occurrences

As it was said before, the registering of occurrences either for normal activities or spontaneous events can be done by voice or typing. The voice feature brings a powerful advantage for vehicular applications, i. e., in case that is the driver who is doing the registering (and not a person besides him/her), he/she just has to press a button and therefore speak naturally to the tablet which handles the rest. Comparing with the typing mode, the voice mode is way more convenient during a driving context. However, the voice mode requires Internet connection since it is used the *startActivityForResult(Intent, int)* method combined with the *Recognizer Intent* and its *ACTION_RECOGNIZE_SPEECH* option [44], which starts an activity that prompts the menu illustrated in figure 3.19. When this menu appears, the user has to speak some (any) words and, after it detects the end of speech automatically, the captured audio signal is sent to Google's servers, translated to text and returned to the *onActivityResult()* method which is called when the above mentioned activity ends. There, the results can be accessed and processed by the application. Similarly to the Bluetooth enabling activity, also in this case the *Main Activity* does the same state changes in its life-cycle: *Resumed*, *Paused* and *Resumed* again. Respecting to the typing mode, it essentially consists in an input text field similar to the figure 3.12.

Contextualizing with the normal operation of the Android application, the registering of occurrences is done as is described in the flowchart of figure 3.20. When the user presses the button, the first thing done is the calculation of the elapsed time, i. e., the difference between the actual moment and the starting time[12] using the *Date* class. After that, it is checked the Wi-Fi connection and, if it is available, it is given a try with the speech recognition. If everything worked well, the result is coupled with the elapsed time and saved in the occurrences file as a new entry. If there is no Internet connection or the speech recognition failed for some reason, the occurrence name is registered by typing in order not to lose any event. A short example of the log file is illustrated in figure 3.18b.

[12]Is the moment when the user presses the *Start Reading* button.

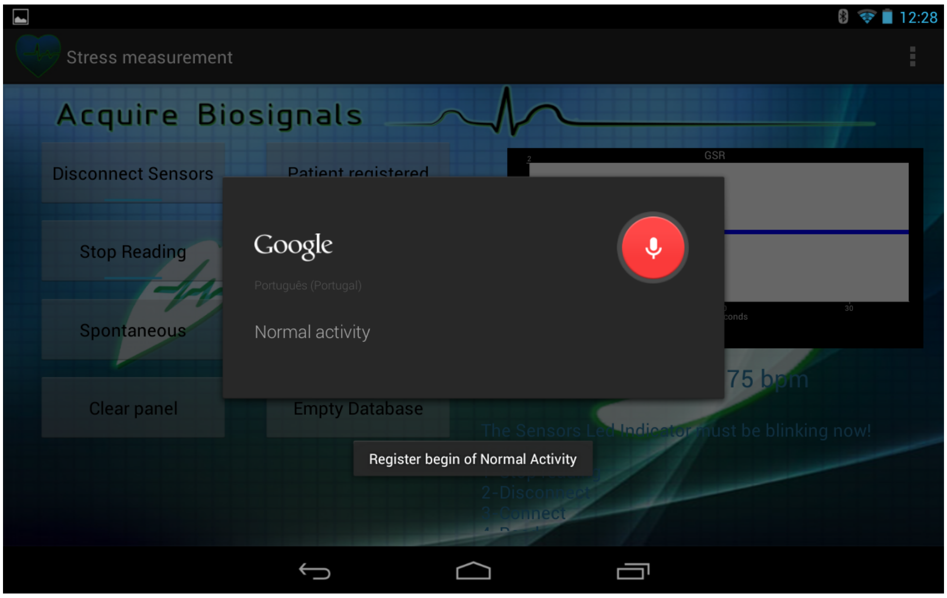


Figure 3.19: Registering an occurrence with speech recognition.

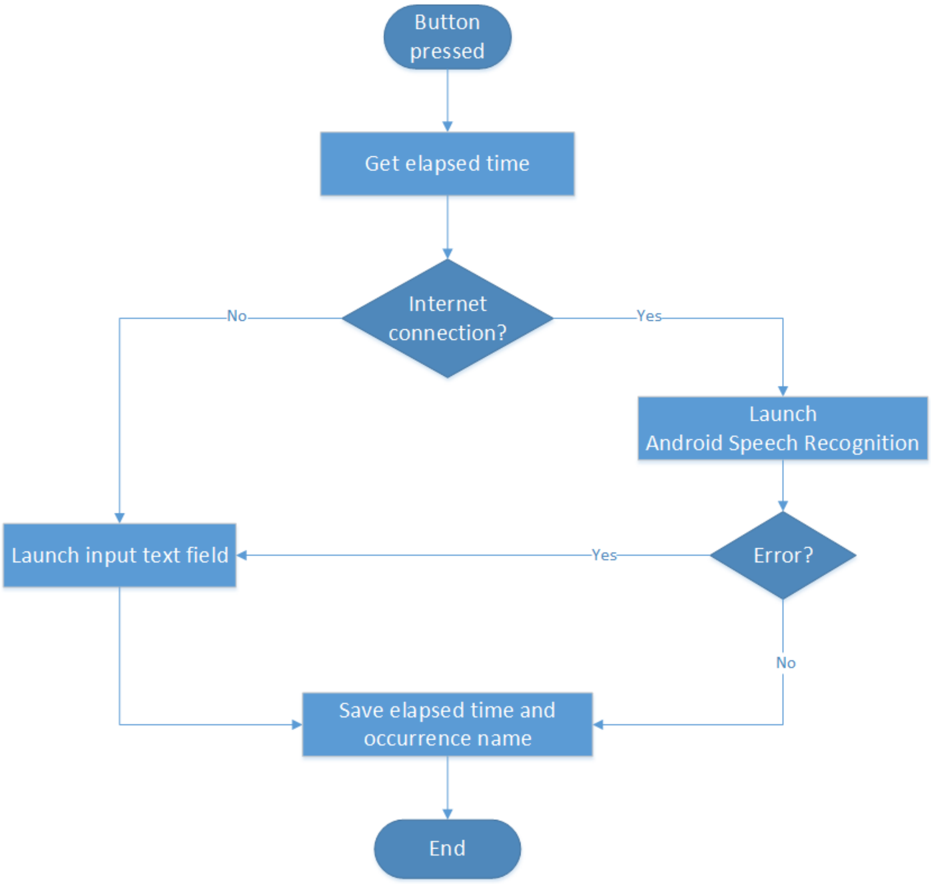


Figure 3.20: Flowchart of the occurrences registering.

3.4.3 Upload data to Dropbox

Dropbox is a service which basically hosts files as a cloud storage and also synchronizes them with a remote computer. In order to transfer all data from the Android device to the Dropbox, it was used their *Core API* [45], which provides methods to do it in a low level control. The first step was to create a folder to be associated with the Android application specifying its name and the *Access Type* permission. In this case, it was chosen the *App Folder* type - which gives access only to the created one - and not the *Full Dropbox* type because it was not necessary to get full access of all user's[13] files and folders. Therefore an *App Key* and an *App Secret* are received to be used in the developer code. In case *Database* is not empty, the next two steps are executed: invoke the *startAuthentication()* method and then write all the files to the specific Dropbox folder. In normal conditions, attending the life-cycle of the *Main Activity* in this segment, it changes its state by this order: *Resumed*, *Paused*, *Stopped*[14], *Started* and finally *Resumed* again - see figure 3.16.

Since the tablet already had the Dropbox application installed, the user has to allow the access to the folder as the figure 3.21 illustrates[15], in this case named *Biosignals*. Either the authentication has succeeded or not, the application returns to the previous activity by calling the *onRestart()* method after the user input, and when the *onResume()* method is called, a new *AsyncTask* is executed in background to upload data. As it can be seen in figure 3.22, all folders (and contained files) from *Database* directory are written to Dropbox in case of a positive authentication. The *Progress Dialog* window indicates how many files are left to be sent in number and percentage. Besides that, it doesn't disappear till all files are uploaded ensuring that the user does not initiate a new experiment.

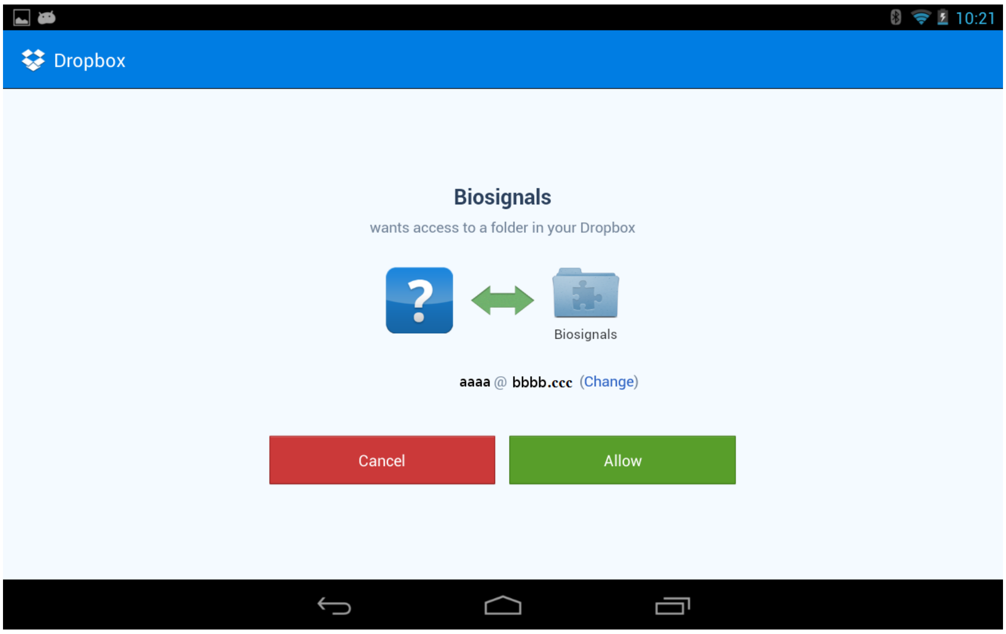


Figure 3.21: First part - authentication

[13] Dropbox user.
[14] The *Main Activity* is hidden (fig. 3.21) and no code from this activity is executed in this state - [37].
[15] In case that Dropbox application is not installed on the Android device, it will browse to the website.

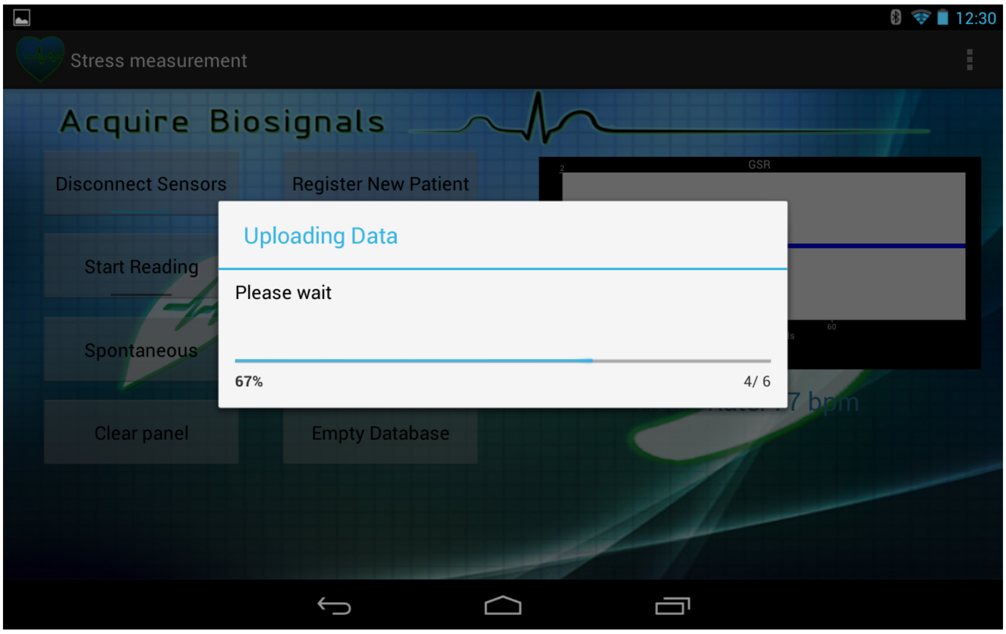


Figure 3.22: Second part - write files

3.4.4 Real-time visualization of biosignals changes

As in all mobile applications, the observation in real-time of any parameters changes gives a better feedback to the user. In this case, it was decided to "follow" the signals coming from the GSR and ECG sensors, more specifically, the skin conductivity and the instantaneous HR. This last parameter is the one calculated by the real-time algorithm - see next subsection.

In order to visualize the SC signal in real time as the graphic plot of figure 3.13 illustrates, it was used the *AChartEngine* software library [46]. Suitable for Android applications, this library is very useful and versatile at the same time because it permits to adjust a lot of parameters as appearance (e. g. colors, axis), chart types (bar, line, pie, dial, etc.) and zoom tools for example. In this case, it was used a line chart type with a time window of 30 seconds. Also from figure 3.13, it can be seen that the HR parameter is shown as an integer value bellow the SC graphic.

The SC and HR values are constantly updated in the UI thread due the action of an *Handler* created for each parameter since they belong to different threads. According to [47], an *Handler* "*is part of the Android system's framework for managing threads*", which basically allows a communication between a background thread and the UI thread by messages processing. The *Handler* usage is necessary because the access to UI objects is not allowed for background threads - in this case for the GSR and ECG threads.

3.4.5 Real-time heart rate algorithm

Currently the ECG sensor sends, via the *Bluetooth* frames, always the value **0** for the instantaneous heart rate because this solution has not been released till this moment. So it would be interesting to develop a real time algorithm able to calculate it. Remembering the

HR definition[16], the algorithm goal is to detect the R peaks and then apply the following formula:

$$Heart\ rate = \frac{60 * f_s}{N_{samples}}(beatsperminute(bpm)) \qquad (3.4)$$

in which f_s is the sampling rate ($250Hz$) of the ECG sensor and the $N_{samples}$ is the number of samples between two consecutive R peaks.

Based in a threshold, the implemented algorithm is essentially divided in two parts and, as all algorithms, it has some advantages and disadvantages which are explained later in 4.3. Illustrated in 3.24a, the first part consists in reading approximately 3 seconds of data and determine the maximum value which will, normally, correspond to a R peak. After that, the important feature (threshold) is calculated as being 60%[17] bellow the maximum value and then, the second part can be started. Till the end of the first part, the heart rate is considered as **0**. Concerning the second part and looking to figure 3.23, it basically consists in counting the number of samples from *Out* to *Out* points, i. e., the ECG blue line. When it detects that the signal is above the threshold (*In* region), it continues to count the number of samples till it recognizes that the signal is lower than the threshold (*Out* region). Only at this last moment, the algorithm executes the formula 3.4 and therefore resets the number of samples to zero - see 3.24b. Till the next calculation, the HR value is maintained since these operations are done sample-by-sample. In normal conditions, the ending of the second part only terminates when the user presses the *Stop Reading* button.

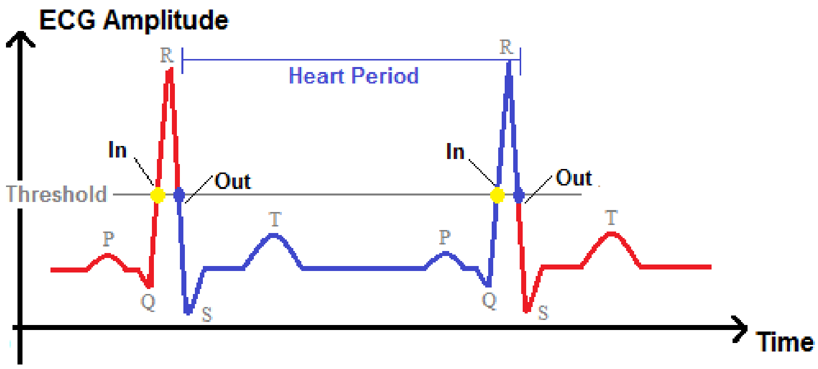
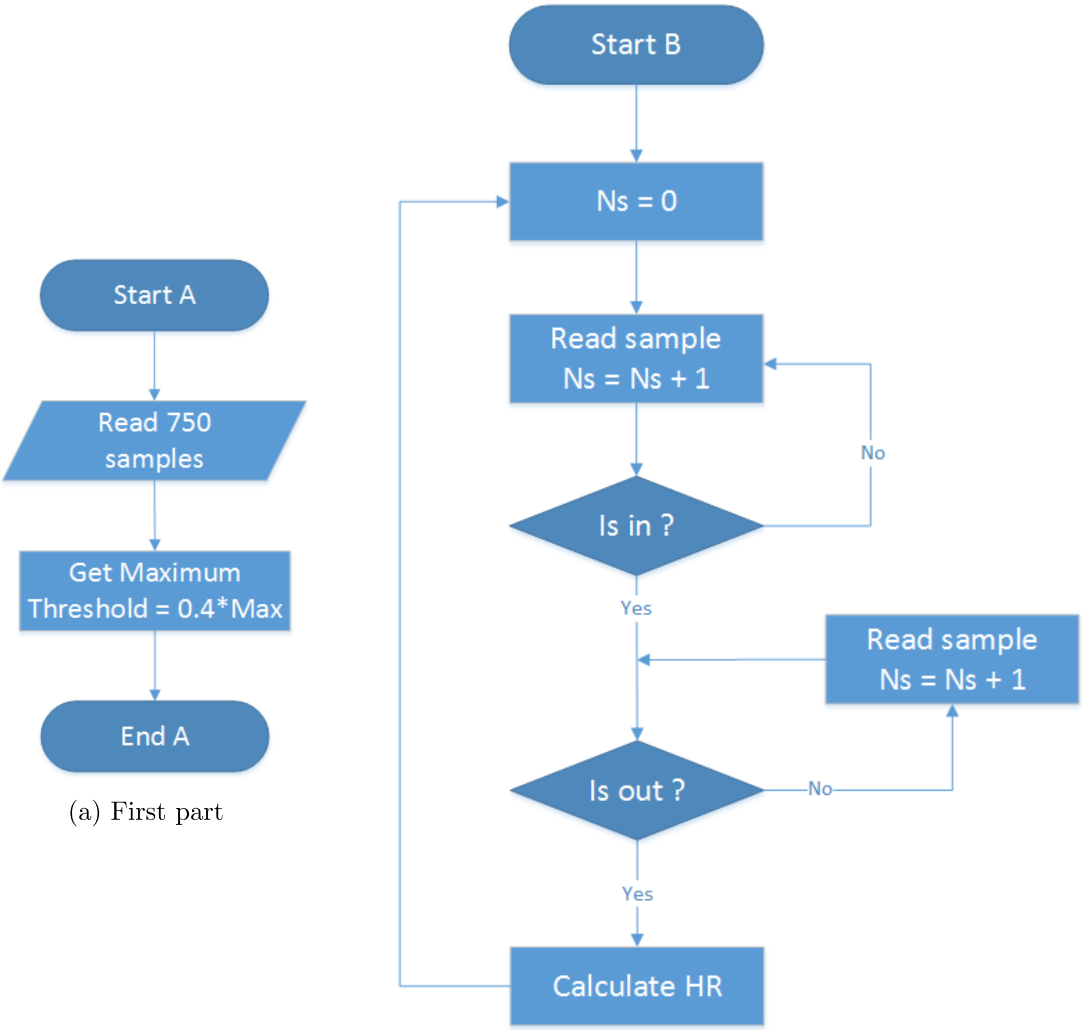


Figure 3.23: HR algorithm applied in the ECG signal.

[16]The heart rate is given by the inverse of the heart period, which is given by the time interval between two consecutive R peaks - [6].

[17]This percentage was found suitable due the observation of some experimental results done in laboratory.



(a) First part

(b) Second part

Figure 3.24: Flowchart of heart rate algorithm.

3.5 Off-line processing with Matlab®

The developed Matlab [24] interface was created to aid the validation of the platform and also to perform a basic off-line processing in order to present experimental results (section 4), serving as the basis for future work. It was developed based in the *GUIDE* tool, which allows to create programs with graphical interface. The figure 3.25 illustrates the global perspective of the interface which consists in five major blocks: importing data, list of occurrences, choose a time interval & amplitude threshold, graphical data and measurements.

This interface is based in algorithms developed by two open source projects: the *Biosig project* [48] and *Ledalab* [49]. Both are open source software libraries under the GNU General Public License. The first one is used to extract features from the ECG signal and the second one to extract features from the GSR signal. The following subsections describe in detail how the interface works and which biosignal features are calculated over a specific time region.

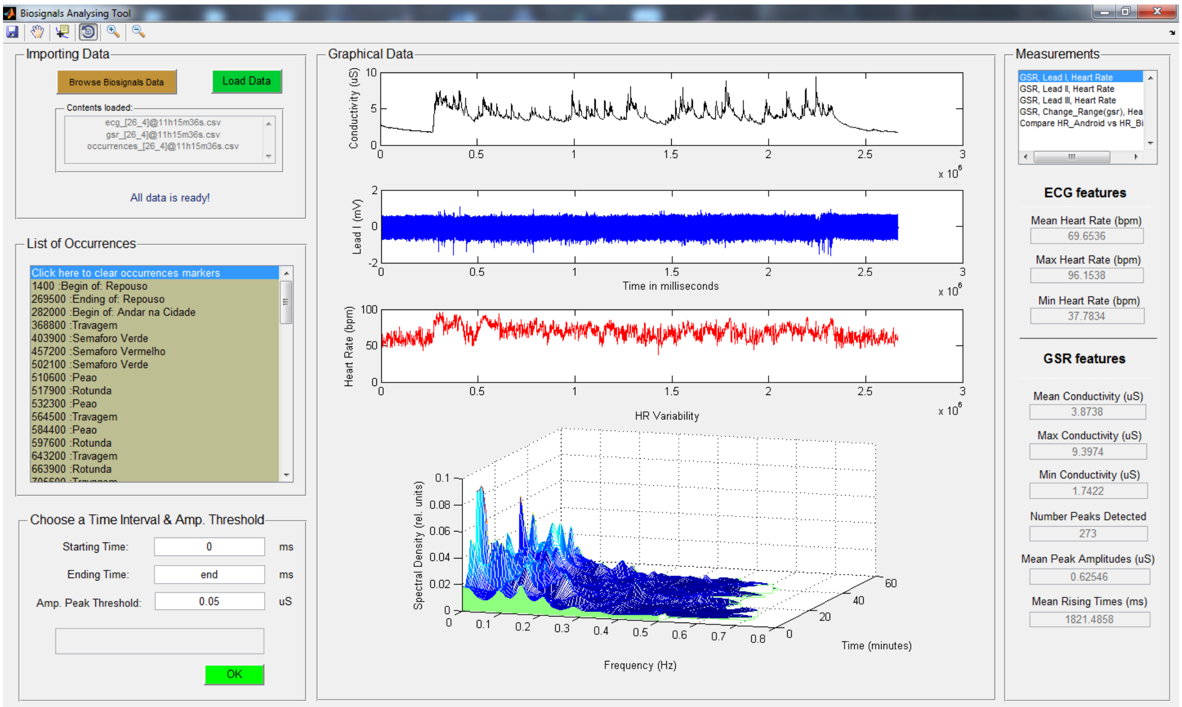


Figure 3.25: Matlab interface developed.

Importing data

The first step in this interface is to load data previously acquired by the Android device. For that, pressing the "*Browse Biosignals Data*" button, it will appear three consecutive windows to load, from any directory, the necessary three different files by this order: *ecg_date.csv*, *gsr_date.csv*, *occurrences_date.csv*. To ensure that all data is load correctly, it will be requested to load a new file if the name of the selected one is not coincident with the expected, i. e., "*ecg_... .csv*", "*gsr_... .csv*" or "*occurrences_... .csv*". After that, the name of the three selected files is displayed as the figure 3.26 illustrates enabling the possibility to verify if the contents to be loaded are in accordance by looking to the dates of all files. If yes, then the user just has to press the "*Load Data*" button and a successful note is displayed if no other error occurred.

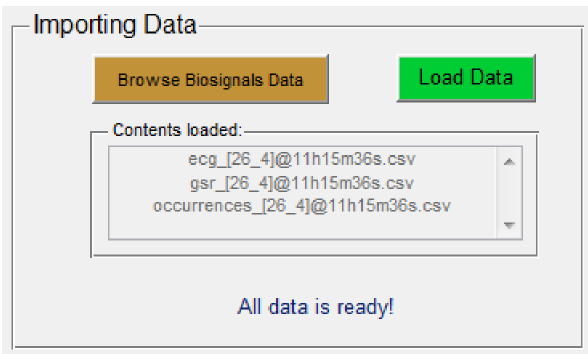


Figure 3.26: Importing data block.

Graphical data

When data is loaded, the graphical data area - four plots - is filled with all temporal data obtained from *ecg* and *gsr* files, i. e., since the start of reading from sensors till the end of it. The bottom graph represents the HRV and it never actualizes its data according to the parameters of the block described after the next one. This *3D* graph illustrates the spectral density of HRV over the frequency range and time - one in one minute. For that, the functions bellow are used and therefore three parameters are extracted from the X struct: $X.ASpectrum$, $X.TotalPower$ and $X.f$.

```
fs = 250;   % sampling frequency of ECG sensor
HDR = qrsdetect(Lead, fs, 2);
X = heartratevariability(HDR);
```

The other three graphic axes can plot different signals depending of the user's wish according to the several options described in the measurements block.

List of occurrences

After loading data, also the list-box shown in 3.27 is actualized with the information contained in the "*occurrences_date.csv*" file. Each line of the list-box has the following format:

Elapsed time (from the beginning) in milliseconds : Occurrence name

Another important feature of this list-box is that, if the user clicks on one of those events, it will be displayed on the top graph a blue vertical line - see examples in chapter 4 - marking the respective event in order to easily associate the occurrence with the biosignals. This vertical line will just appear if the chosen occurrence is within the temporal range defined in the block described next. To clear all previously selected occurrences it is necessary to click in the first line of the list-box.

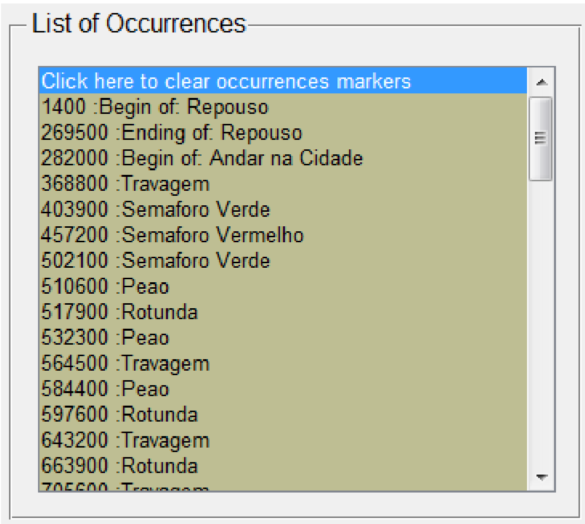


Figure 3.27: List of occurrences block.

Choose a time interval & amplitude threshold

As the figure 3.28 illustrates, this block is intended to choose a time interval and the amplitude threshold. The first one is to restrict acquired data to a specific temporal range and basically it has two parameters: a starting time and an ending time, which can be chosen by looking to the list of occurrences. The second one is to select only GSR peaks with an amplitude difference above this threshold, i. e. the difference between the peak and the onset amplitudes, turning this versatile option useful to detect only huge peaks. After the parameters have been chosen, if the *OK* button is pressed and no error occurs as numeric type or starting time being greater than ending time, the graphical data area will be actualized as well as the GSR and ECG features - see next block. This block is useful to better analyse and observe the biosignals evolution during specific normal activities as driving in a city.

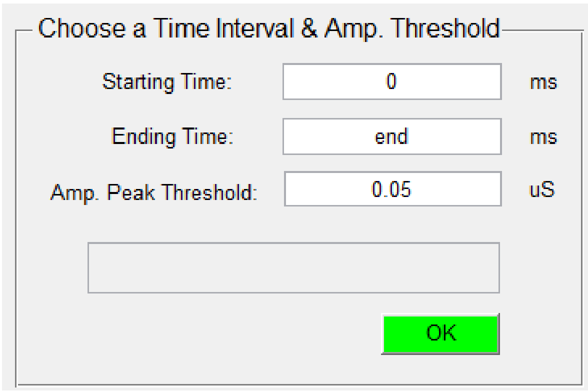


Figure 3.28: Temporal and threshold parameters block. Note: the 'end' word just appears by default in the initialization of the program.

Measurements

The measurement block consists in a list-box and fields with GSR and ECG features. In these fields, according to the time interval and amplitude threshold previously chosen, the mean, maximum and minimum values of SC and HR are showed. Also for GSR, it is shown the number of peaks detected as well as the mean of their rising times and relative peak amplitude.

Concerning the list-box, the user can selected what kind of data wants to visualize. So there are five options and each of them has three fields which represent, by order, the correspondent plot. These options are enunciated bellow and the first one is the default, i. e., after load data, the plots illustrated are those from the first option.

- GSR, Lead I, Heart rate
- GSR, Lead II, Heart rate
- GSR, Lead III, Heart rate
- GSR, Change range (GSR sensor), Heart rate
- Compare HR Android and HR *BioSig*

The fourth option is basically to check the moments of when the GSR sensor changed the range in order to calculate the skin conductivity. The last option was created to characterize the quality of real-time HR algorithm developed. As the ECG sensor does not send it, neither the Android device calculates it, the Lead I vector is calculated here by software as the difference between Lead II and Lead III. The HR vector from the first four options is calculated always based in Lead II and in *nqrsdetect* function from *BioSig*. The Matlab code bellow shows how the heart rate vector is calculated in bpm.

```
fs = 250;    % sampling frequency of ECG sensor
qrs_indexs = nqrsdetect(Lead, fs);
qrs_indexs_diff = diff(qrs_indexs);
HR_features.HR_vector = 60*fs./qrs_indexs_diff;
```

Concerning the GSR features, specifically for the number of peaks and their onset and rising times, it is used the *smooth_adapt* and *trough2peak_analysis*. The last one, which calls the *get_peaks* function, was slightly adapted according to the interface needs.

Chapter 4

Experimental results

4.1 Summary

This chapter aims to validate the platform, make a comparison between the developed HR algorithm and the one from the *Biosig* project and also exhibits results from real experiments: vehicular navigation, video game and tilt table. To conclude, some remarks about the measured biosignals and registered occurrences are mentioned.

4.2 Platform validation

In order to validate the platform, four main experiments were done: the first was to check if it acquires data and registers occurrences properly; the second one to verify its reliability (no loss of data) for long time experiments which was done after confirming the adequate operation of the first test; the third one to check lifetime of sensors battery; and finally the last one to estimate the phase error committed when it is marked the starting time of an experiment.

First test - verify the normal operation

So the first test consisted in simulating some occurrences from both types (normal activities and spontaneous events) at specific timings. The ECG sensor was normally placed in the body but not the GSR sensor because since it is easy to create pulse responses by pull-in and pull-out fingers of the electrodes due to SC properties, one pulse with a small duration was triggered this way for each occurrence in order to check the synchronism between the biosignal and the occurrences. In this test, it was followed the procedure described by the table 4.1 [1]. The *Wi-Fi* was turned off before start reading from sensors in order to register the occurrences 'a', 'b' and 'c' by typing. After ending the activity 'b', the *Wi-Fi* was turned on to enable the voice registering mode in order to say the portuguese word *'experiência'* which means experiment. The experiment was stopped fifteen seconds after registering this last occurrence.

Results of the first test are illustrated in figure 4.1 in which the ECG signal is not present due to the high amount of samples acquired for fifteen minutes, but, has it will be shown in following results, the signal sent by Schimmer sensor is well processed by the platform

[1] N/A means stop reading from wireless sensors which is not registered as an occurrence.

Elapsed Time(m:s)	Elapsed Time(ms)	Occurrence	Registering mode
1:00	60000	Spontaneous: 'a'	Typing
3:30	210000	Begin Activity: 'b'	Typing
7:00	420000	Spontaneous: 'c'	Typing
12:45	765000	Ending Activity: 'b'	Typing
15:00	900000	Spontaneous: 'experiência'	Voice
15:15	915000	N/A	N/A

Table 4.1: First experiment procedure to validate the platform.

processing unit. Looking at the HR Android curve (bottom), the HR values are within the expected range and it can also be verified, at the first seconds, the transition between the first and second part of the implemented HR algorithm - section 3.4.5. The quality of this algorithm is studied in more detail in section 4.3. Comparing the elapsed times that are in the list box with those of table 4.1, it can be concluded that they coincide although a small error is verified due to the delayed human reaction when pressing the button. Looking at the top graphic (SC), it is observed that the synchronism pulse-occurrence is well performed although there exists a delay observed in the zoomed area associated to the previous source origin. This time delay for each step was approximately $500ms$. The elapsed time of the last SC sample was approximately $915332ms$ and for the last ECG sample was approximately $915453ms$ which are coherent with the expected value if the configured sampling rates and the time delay of pressing the button are taking in mind. Analysing the name of registered occurrences, it is concluded that all of them were well marked specially the last one since it was done by voice. Doing other experiments saying random words and phrases, it was concluded that the speech recognition gives good results and therefore is a valuable tool to be applied in this context.

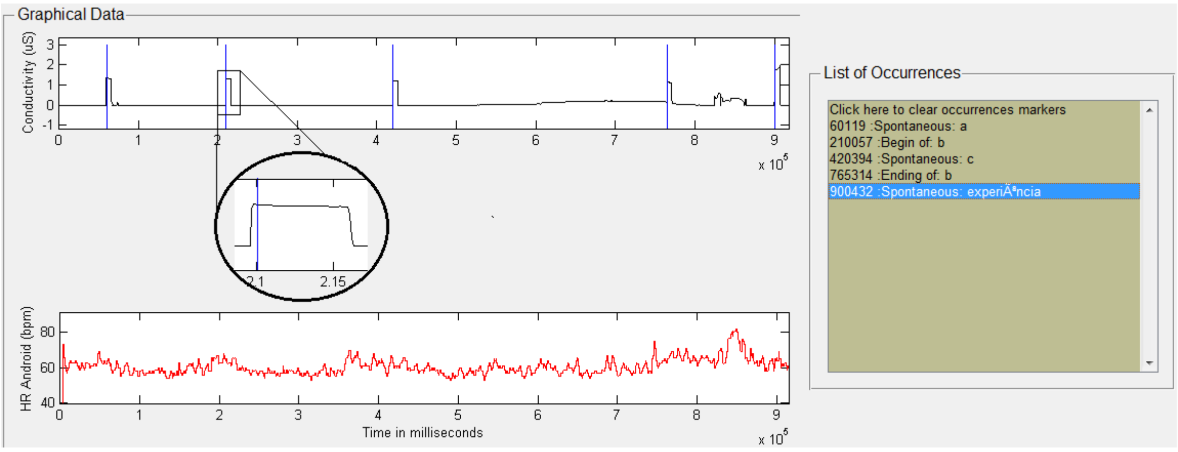


Figure 4.1: First experiment procedure to validate the platform.

Second test - reliability for long time experiments

Regarding the second test, it was to check how much time one experience could take, so both sensors and tablet were fully charged and a new experiment was started. Always with

Wi-Fi off and no other applications running in parallel, some occurrences were marked to check again the same issues done in the first test and it was obtained the intended synchronism as before. The test started at *15:27* and it has to be terminated at 23:38 of the same day since it was the moment when the tablet only had only 10% of battery left. The elapsed time of the last sample coming from GSR sensor was 29459765.625ms and from the ECG sensor was 29460046.875ms which are coherent with the 8 hours and 11 minutes. The results of equations 4.1 and 4.2 prove that no data frame was lost during the experience for GSR and ECG sensors respectively[2] since they are equal to the amount of received samples.

$$N_{samples} = \frac{Last\ time(s) * 32768}{3200} + 1 = 301669 \tag{4.1}$$

$$N_{samples} = \frac{Last\ time(s) * 32768}{128} + 1 = 7541773 \tag{4.2}$$

So it can be said that for 'outdoor' experiments the system can properly acquire data only during a bit more than 8 hours approximately. However, the total size of the three created files was 287MB and some problems occurred in the Matlab interface related with memory[3].

Third test - sensors battery lifetime

About the third test, a new experiment was started with both sensors fully charged, tablet plugged to the power source and the other same conditions of the second test. As expected, due to the higher sampling rate, the ECG sensor was the first stop transmitting data. Its last sample had a elapsed time of 47493609.375ms, which means that the system worked approximately during 13 hours. Once again, no data frames were lost during this test, however, by observing the SC graphic, it was noticed some stuttering - issue not verified in the second test - right before ending the application, i. e. after ECG sensor turned off by itself, maybe due to some problems related with memory used for the graphic visualization or even with the threads managing.

Fourth test - estimate the initial phase error

Since the Android device has to request data from two different sensors by sending the *Start Streaming Command*, it automatically implies an error when determining the initial instant of the experiment. As this phase error affects the elapsed times of the registered occurrences, an estimative was made to calculate it - figure 4.2. Hence, it was verified that this error was always less or equal than 1 millisecond which is not critical for the ambit of this project.

[2]The meaning of values 32768, 3200 and 128 are explained in section 3.3.1.

[3]It was used a 32-bit Matlab version (R2012b).

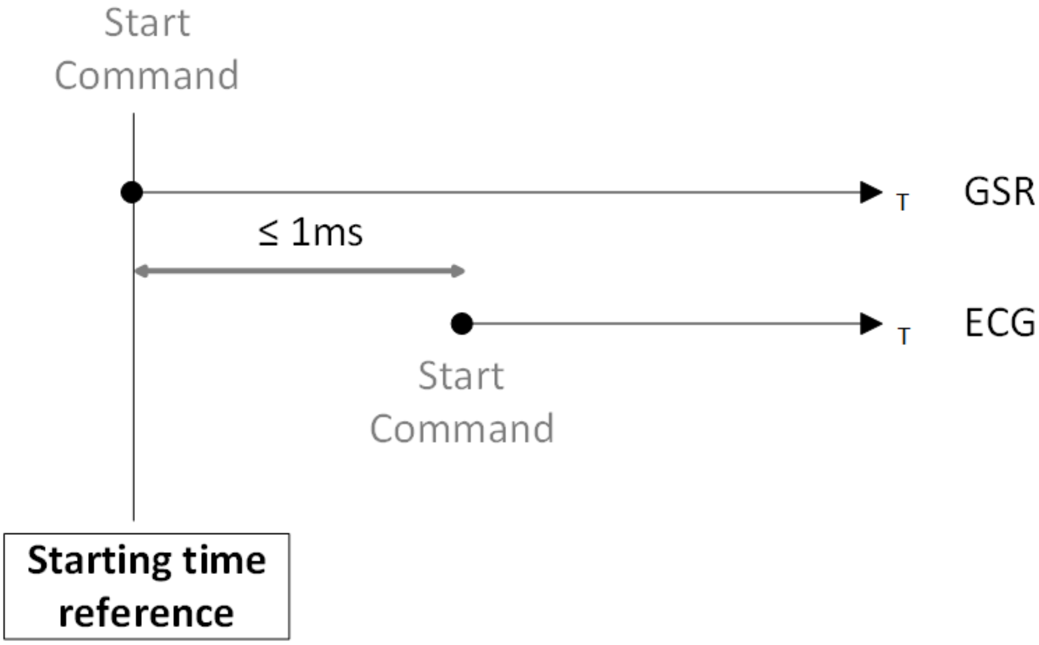


Figure 4.2: Error made in determining the initial instant - phase error.

Other verifications

It was also confirmed in other experiments that the Android application gives the possibility to register occurrences by typing in case that there is no internet connection or if the translate fails, so there is no loss of occurrences. Moreover, the uploading to Dropbox of all patients directories from the *Database* directory was always successfully done - as is logic, the time to upload them depends on their size.

It was verified along several experiments that sometimes it takes too much time to connect with both sensors. The *AsyncTask* used for this purpose was to implement a loop code to guarantee that after pressing the *Connect Sensors* button, the user only can interact again with the Android application after the connection is really established[4].

Just to conclude this validation, also the action of the Android application button "Empty Database" works well.

4.3 Comparison of heart rate algorithms

In this section, the quality of the heart rate algorithm implemented in the processing unit is analysed. Starting with figure 4.3, three different graphics are illustrated: ECG(Lead II), HR calculated in real-time by the Android device and the HR calculated off-line in the Matlab interface. The calculation of the two HR vectors are always based in the *Lead II* vector. Just to remember in the Android case, the HR value is calculated beat-to-beat, i. e., while the next R peak is not found, the previous HR value is maintained. Comparing HR vectors, it can be concluded that the real-time algorithm is working well - in fact, this is verified in the majority of all acquired data.

[4] The is due the fact that the *connect()* method blocks till the connection succeeds or fails.

However, it has imperfections that occur rarely and some of them were detected which are described next. Remembering that the algorithm is based in a threshold defined as 60% bellow the R peak, two imperfection types can be seen in figures 4.4 and 4.5. In the first one, from 865 to 880 seconds, the baseline[5] oscillates and this is due to subject movements that affect the potential difference measured by electrodes. Therefore, since the threshold is bellow that *noise* value visible in the zoomed area, the algorithm considered it as a R peak and then calculated a high HR value because the number of samples counted within the RR interval decreased significantly - see formula 3.4. In the second one, it is seen that the T peaks have the same amplitude than the R peaks, so the failure reason is the same of the previous imperfection but from a different origin[6] - compare with figure 3.23. So the worst case for the real-time algorithm is this last one because it constantly gives wrong values since the ECG wave maintain that pattern and the first one only gives wrong values sporadically.

Nevertheless, the bottom curve of figures 4.4 and 4.5 indicates that *BioSig* functions apply algorithms more robust to handle several artefacts. Since one of the aims of this project was to acquire data to posteriorly analyse it off-line, it is concluded that the usage of the *Biosig* project is very reliable and useful.

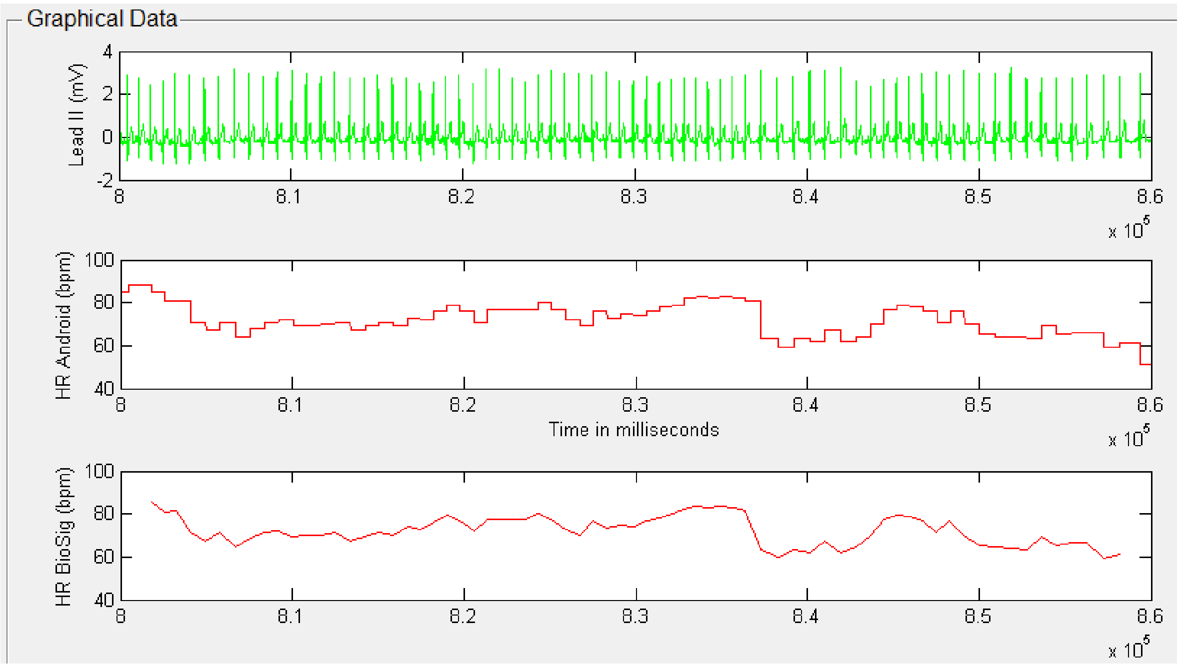


Figure 4.3: Real-time heart rate algorithm - no fails.

[5]In this case, the baseline is the $0mV$ line.

[6]The reason that explains this (non-expected) high value of T peak is related with the ventricular repolarization (relaxation) [6]

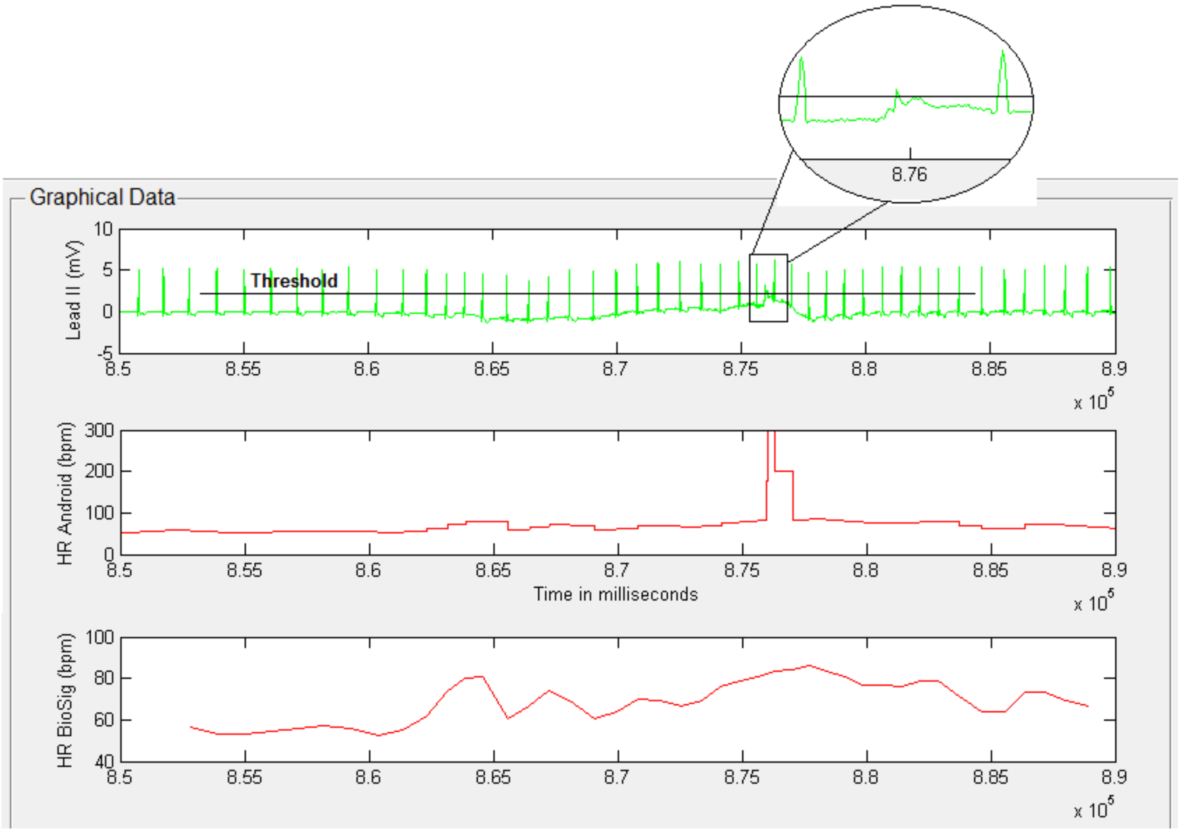


Figure 4.4: Real-time heart rate algorithm - imperfection 1.

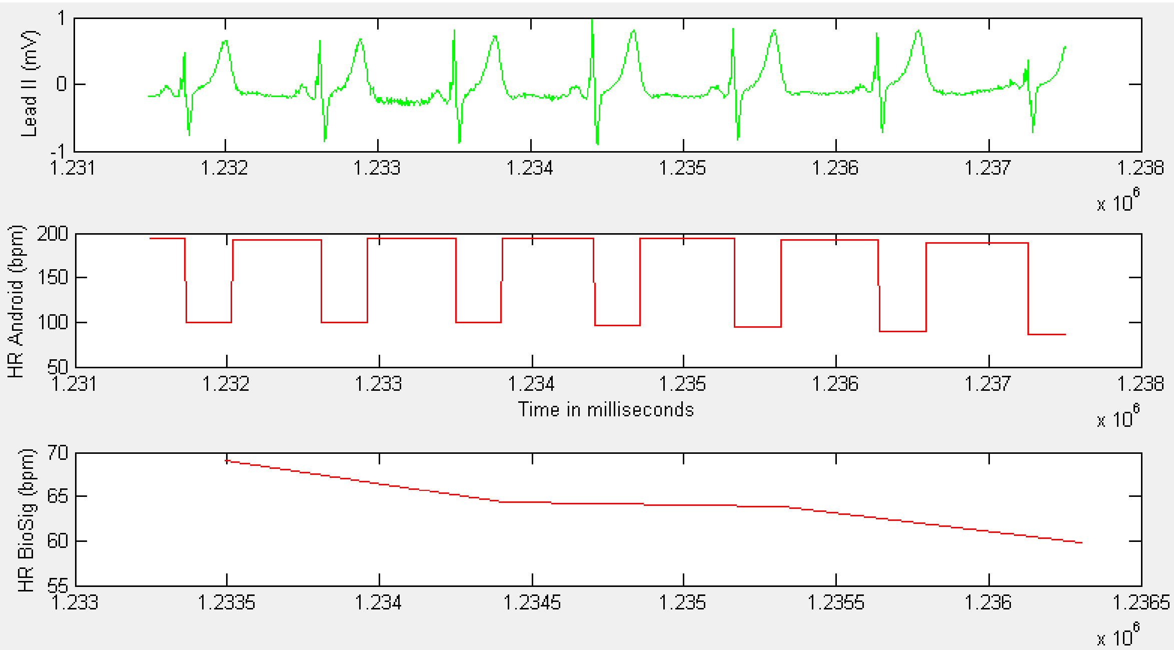


Figure 4.5: Real-time heart rate algorithm - imperfection 2.

4.4 Vehicular navigation context

In this context, real car driving experiments were done in the city of Aveiro with four subjects, all of them in different days, with the same routine, starting approximately at $11h00$ *am* with a duration of nearly 45 minutes. The experimental procedure was very simple: at the beginning the driver stayed five minutes in the car relaxing and trying to get used to the car; then he/she started to drive in the city for some minutes; the next step was to drive in a highway in direction to the beach named Barra, return back and finally do the inverse route, i. e. driving in the city and then park the car. The experiment was only finished after five minutes of relaxing seated in the car. The second city driving phase was shorter (about 2 minutes) when compared with the first. During all these phases, spontaneous events as traffic lights changes, pedestrian crossing road suddenly, passing roundabouts were marked by a co-pilot. At the time of these experiments, it was used an Android smartphone in which it was not possible to get an Internet connection. Due that, the register of events had to be done always by typing, thus, not to put in risk the driver's safety, the aid of the co-pilot was necessary.

The first subject was a 25 years man, the second was a 50 years woman, the third was a 23 years man and the last one was a 37 years man. The results of this mental-physical activity are presented next with a brief analysis for each subject. The error presented in some table cells means that the maximum HR value was affected by '*noise*' or body movements (artefacts). Theoretically, it is expected that the stress level of persons is higher when driving in the city, lower in resting phases and intermediate in the highway. A small note for the peaks detected in the SC signal, which were chosen to be above a threshold of $0.1uS$. Concerning yet these peaks, the peak amplitude and rising times are measured since the onset and the maximum value of the peak.

Results from subject 1

The results of this masculine subject are presented in table 4.2 and figure 4.6.

——	Rest 1	City 1	Highway	City 2	Rest 2
Mean HR (*bpm*)	59.8	74.6	67.8	72.3	61.4
Maximum HR (*bpm*)	76.1	96.2	88.2	89.8	84.8
Minimum HR (*bpm*)	43.1	48.5	37.8	53.6	46.3
Mean Conductivity (μS)	2.1	4.4	4.2	4.8	2.4
Maximum Conductivity (μS)	2.9	7.5	8.9	9.4	4.8
Minimum Conductivity (μS)	1.8	3.2	2.8	3.5	1.8
Number Peaks Detected	1	97	77	34	3
Mean Peaks Amplitude (μS)	0.12	0.74	0.79	0.85	0.14
Mean Peaks Rising Time (s)	3.2	2.1	1.9	1.6	2.2

Table 4.2: Driving results of subject 1 - 25 years old man.

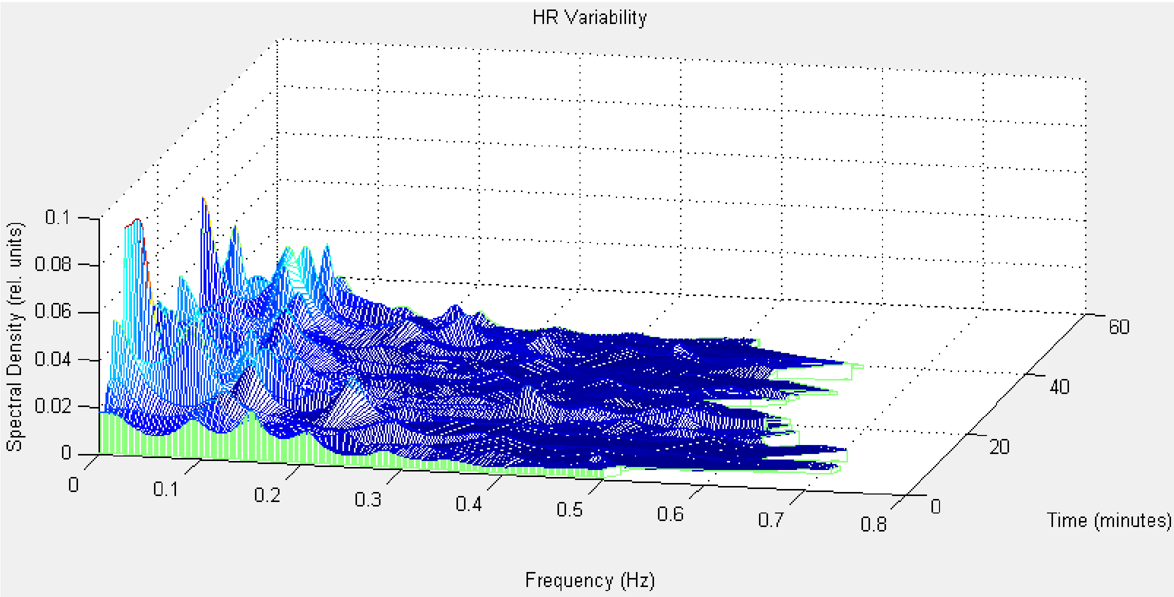


Figure 4.6: HRV of a 25 years old man - driving context.

The HR mean value is higher when the subject is in city phases. The maximum and minimum HR happen during *City 1* and *Highway* phases respectively. The mean SC behaved like the HR, i. e., it raises from *Rest 1* to *City 1*, decreases in the *Highway*, raises again in *City 2* and it drops significantly in *Rest 2*. The maximum and minimum SC happen during *City 2* and *Rest 1/2*. Although the number of detected peaks depends on the duration of each phase, it was *City 1* which presented more. In the resting phases, is verified that the mean amplitude is lower and the mean rising time is higher. Looking at the spectral density of HRV along all the experiment, the low frequencies dominate which are related with the SNS.

Results from subject 2

The results of this female subject are presented in table 4.3 and figure 4.7.

——	Rest 1	City 1	Highway	City 2	Rest 2
Mean HR (*bpm*)	67.9	69.7	66.6	69.6	64.3
Maximum HR (*bpm*)	74.6	144.2 (error)	75.4	79.4	80.2
Minimum HR (*bpm*)	60.2	57.3	56.4	58.1	56.4
Mean Conductivity (μS)	9.0	11.5	11.9	12.3	10.8
Maximum Conductivity (μS)	11.6	14.9	15.0	14.4	13.7
Minimum Conductivity (μS)	7.4	9.2	9.4	11.3	8.8
Number Peaks Detected	52	308	70	50	41
Mean Peaks Amplitude (μS)	0.55	0.69	0.83	0.68	0.66
Mean Peaks Rising Time (s)	1.8	1.6	1.9	1.5	1.9

Table 4.3: Driving results of subject 2 - 50 years old woman.

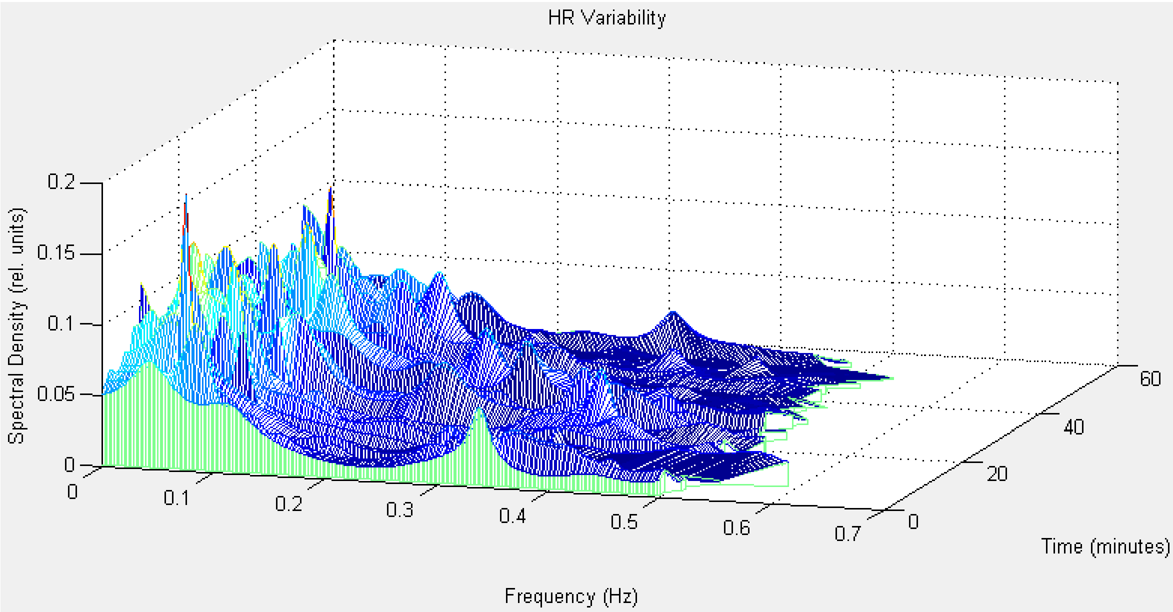


Figure 4.7: HRV of a 50 years old woman - driving context.

Once again, the HR mean is higher for the both *City* phases. The minimum HR occurred in the *Highway* and *Rest 2* phases. Here, the mean conductivity didn't follow the HR behaviour since it was constantly increasing till the *City 2* phase and then decreased in the resting phase. The maximum and minimum SC values happened during *Highway* and *Rest 1* phases respectively. It has to be highlighted the number of detected peaks during *City 1* comparing with others phases. The mean of amplitudes was higher in the *Highway* phase as well as the mean rising times.

Results from subject 3

The results of this masculine subject are presented in table 4.4 and figure 4.8.

—	Rest 1	City 1	Highway	City 2	Rest 2
Mean HR (*bpm*)	58.6	63.3	60.1	62.5	59.0
Maximum HR (*bpm*)	72.8	172.4 (error)	142.9(error)	75.0	75.0
Minimum HR (*bpm*)	52.3	54.5	51.7	54.4	49.2
Mean Conductivity (μS)	1.0	2.0	2.7	3.4	3.4
Maximum Conductivity (μS)	1.3	2.9	4.4	4.2	5.5
Minimum Conductivity (μS)	0.7	1.2	2.0	2.8	2.5
Number Peaks Detected	4	32	29	7	44
Mean Peaks Amplitude (μS)	0.21	0.24	0.47	0.50	0.55
Mean Peaks Rising Time (s)	4.2	3.2	3.6	2.5	2.4

Table 4.4: Driving results of subject 3 - 23 years old man.

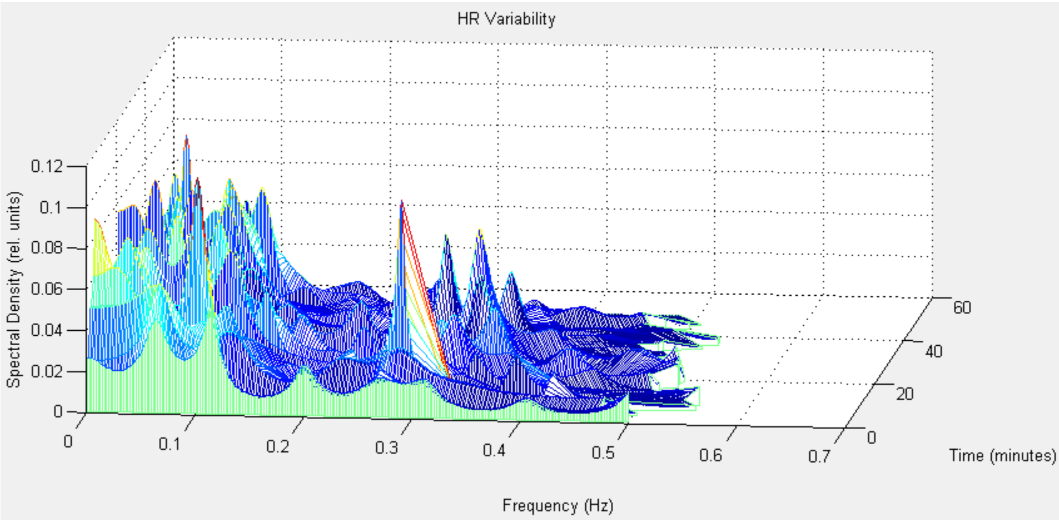


Figure 4.8: HRV of a 23 years old man - driving context.

The HR mean follows the theoretical behaviour, i. e., raise, decrease, raise and decrease again. The minimum HR happened in *Rest 2*. The mean SC was constantly raising and from *City 2* to *Rest 2* it maintained. The maximum and minimum SC happened in *Rest 2* and *Rest 1* respectively. The higher number of detected peaks in *Rest 2* is maybe due his mental activity which can increase the stress level depending on what it consists. The mean of amplitudes constantly increased till the end of the experiment and the mean of their rising times started with a long duration followed by a decrease, then an increase again and therefore it decreases till the last phase. Analysing the HRV, it is clear that the 'activation' of the PNS (high frequencies) occurred often during the experiment comparing with the other subjects.

Results from subject 4

The results of this masculine subject are presented in table 4.5 and figure 4.9.

—	Rest 1	City 1	Highway	City 2	Rest 2
Mean HR (*bpm*)	63.7	67.7	65.9	68.2	61.6
Maximum HR (*bpm*)	80.2	91.5	85.2	90.4	79.4
Minimum HR (*bpm*)	45.7	40.7	42.5	43.2	41.3
Mean Conductivity (μS)	1.2	1.8	2.3	2.8	2.6
Maximum Conductivity (μS)	1.7	2.5	3.0	3.8	3.4
Minimum Conductivity (μS)	1.0	0.9	1.9	2.3	2.3
Number Peaks Detected	2	53	44	36	10
Mean Peaks Amplitude (μS)	0.17	0.19	0.22	0.21	0.16
Mean Peaks Rising Time (s)	2.1	2.3	2.6	2.0	2.2

Table 4.5: Driving results of subject 4 - 37 years old man.

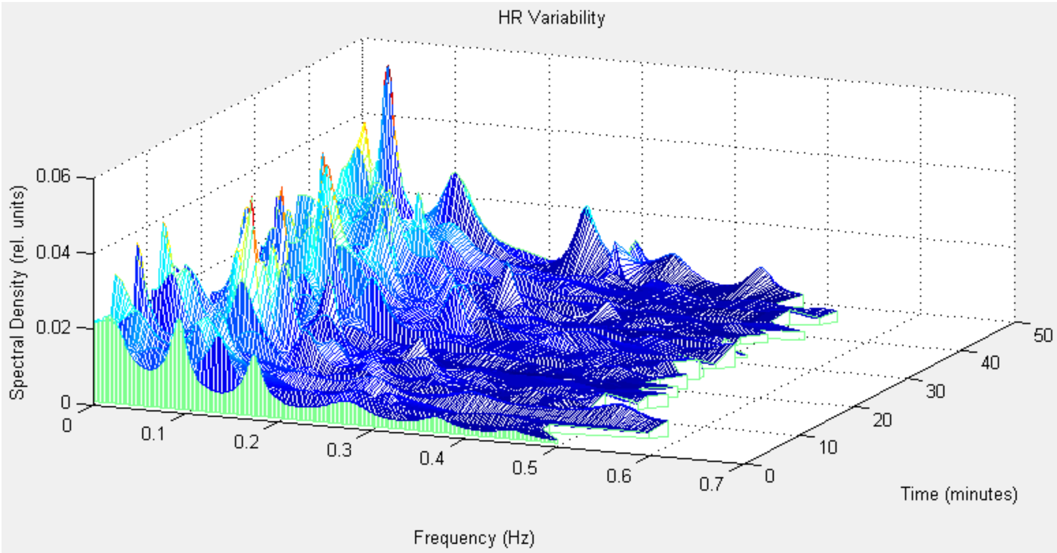


Figure 4.9: HRV of a 37 years old man - driving context.

Once again, the HR mean value followed the theoretical behaviour. The maximum and minimum HR occurred both during the *City 1* phase. The SC raised till *City 2* and then decreased in the last phase. The maximum and minimum SC occurred in *City 2* and *City 1* respectively. The number of detected peaks is lower in the resting phases and it was higher during *City 1*. Concerning the mean amplitude of peaks, it was higher during the *Highway* as well as their mean rising time. Analysing the HRV, it can be noticed an activation of the PNS at the end of the experiment - resting phase.

Comparison among all subjects

In order to make a comparison among all subjects, the table 4.6 presents, for each driver, the values of the entire experience: mean of HR, total number of detected peaks, mean amplitude and rising times of all peaks.

——	Driver 1	Driver 2	Driver 3	Driver 4
Mean HR (*bpm*)	67.2	67.6	60.7	65.4
Total Number of Peaks	212	521	116	145
Mean All Peaks Amplitude (μS)	0.528	0.682	0.394	0.190
Mean All Peaks Rising Time (s)	2.20	1.74	3.18	2.24

Table 4.6: Mean values of detected GSR peaks for all road circuit.

It is concluded that drivers 1 and 2 present higher heart-rate mean values, total number of peaks as well as mean amplitude of peaks. On the other hand, drivers 3 and 4 present the higher rising times, which means that the adaptation to stress responses differs from person to person.

Other results

Next, it is shown the variation of SC according to some (spontaneous) occurrences that were happening during driving. In figure 4.10 there are two occurrences: green traffic-light and pedestrian crossing the road. In figure 4.11, three occurrences are presented: green traffic-light, roundabout and a crossing with another car.

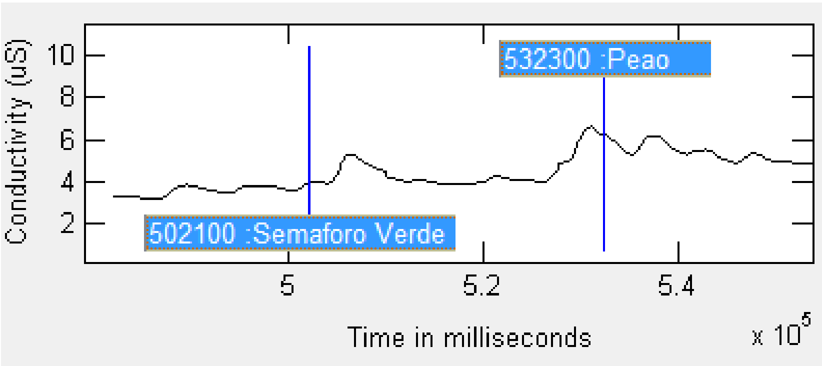


Figure 4.10: Association occurrences-biosignal (1).

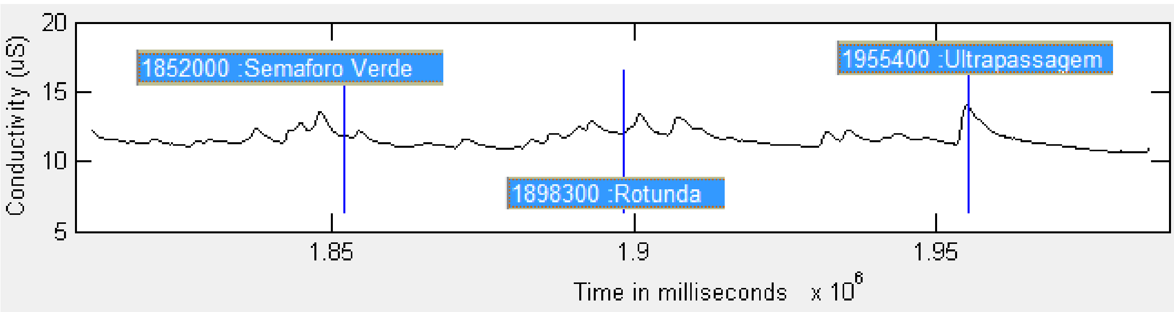


Figure 4.11: Association occurrences-biosignal (2).

It is observed that the GSR signal can identify relevant events and also that some of them cause more *stress*. As an example, it can be seen that the peaks amplitude associated to the pedestrian crossing the road and the car crossing are clearly higher than the others.

4.5 Video game context

In this context, one subject was evaluated when playing a video game which requires a high mental activity. The game consists in embodying the role of a person that is lost within a forest during night and has to find a specific number of special pieces that are hidden. However, this task is influenced by the existence of a special character that can appear at any time in front of the main character to scare him/her. This spontaneous *stressor* is also supported by scary background music which augments its intensity after picking up each piece. So the goal is to pick up the pieces without being caught by the bad guy. During the experiment, the registering of occurrences was done by other person. As the previous context, it was given five minutes to rest before and after the real playing. A small note for the peaks detected in the SC signal, which were chosen to be above a threshold of $0.1uS$.

Table 4.7 presents a resume of all the experiment, the figure 4.12 illustrates the behaviour of SC and HR signals during playing phase, and table 4.8 enunciates the registered occurrences.

——	Rest 1	Playing	Rest 2
Mean HR (bpm)	58.6	62.3	57.8
Mean Conductivity (μS)	1.8	2.2	2.1
Number Peaks Detected	7	31	3
Mean Peaks Amplitude (μS)	0.27	0.33	0.19
Mean Peaks Rising Time (s)	5.0	4.2	3.4

Table 4.7: Results of subject 1 - video game context.

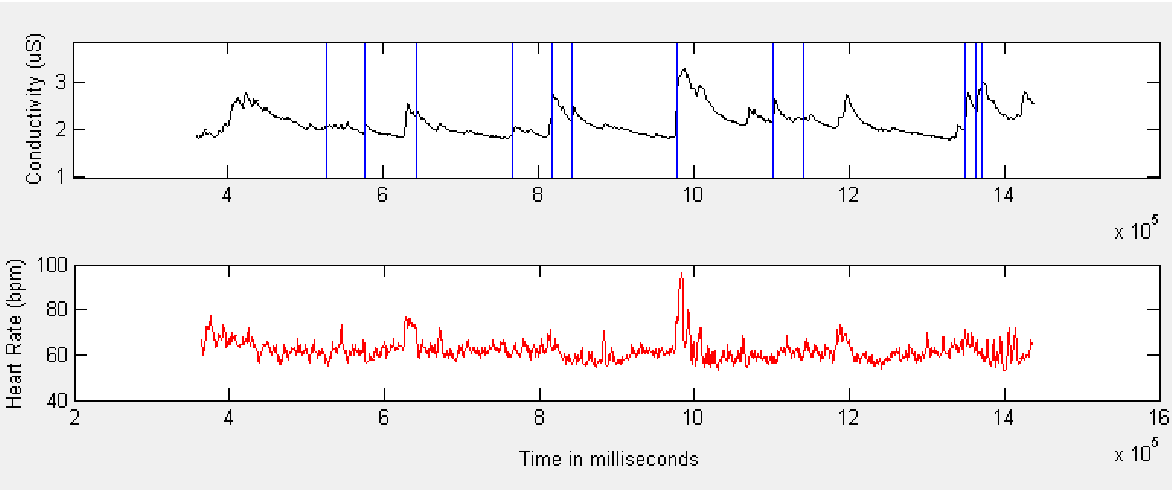


Figure 4.12: SC and HR of subject 1 - during playing phase.

Elapsed time (ms)	528976	576658	644187	766360	818500	844014
Occurrence	piece	fright	fright	piece	fright	fright
Elapsed time (ms)	978320	1101244	1140412	1347646	1362860	1369881
Occurrence	relax	fright	piece	fright	fright	end

Table 4.8: Occurrences and their elapsed time during the play phase.

It is verified that, during the play, the mean HR, SC, number of detected peaks and peaks amplitude was higher than in resting phases. Every time that the bad guy appeared, it originated a peak in the SC. But not only this was the reason to originate peaks, i. e., the peak detected before the 1000 seconds is related with a deep respiration that the subject did in order to relax himself. This peak had an amplitude of $1.45uS$ and a rising time of 10.9 seconds. There it is also observed the significant increase of the HR to almost $100bpm$.

4.6 Tilt table context

In an occasional conversation with cardiologists of the nearest hospital of our laboratory, it was discussed a few thoughts about the purpose of this project and how it could be perhaps extended to new diagnostic methods in the cardiology branch. It was pointed that our platform acquires two biosignals (ECG and SC) that carry reliable information about the human stress and how this information is related with the SNS and PNS. In doctors perspective, it caught their attention due the fact that some diagnosis of cardiac diseases are supported by measured parameters which reveals the activation or not of these two systems. For example, a well-known characteristic of high-risk patients after myocardial infarction is the increase of the sympathetic activity and, in parallel, the reduction of the parasympathetic activity [50]. Some of those parameters are derived from the ECG signal as the HR and HRV. However, till now, the GSR signal is not approved among the medical community has being a marker of the SNS (de)activation, so it would be interesting to research in deep the characteristics of this biosignal and observe if it can be properly included in future diagnosis.

Here comes the tilt table which is no more than a table that can be tilted. This instrument is used by the cardiologists in some diagnosis, e. g., syncope, which is getting more common nowadays. According to [50], syncope is defined "*as a transient, self-limiting loss of consciousness caused by global cerebral hypoperfusion, accompanied by a loss of voluntary muscle tone as well as spontaneous recovery when lying supine*".

Thereupon, combining a tilt table test with impedance cardiography, non-invasive blood pressure and continuous ECG measurements - which is what cardiologists do in their labs - enables physicians to give a detailed diagnostic about the ANS which includes the SNS and PNS. Independently of diagnosis type that is intended to do, typical tilt table testing protocols consist in several phases such as applying sensors, resting phases in supine position, tilt the table with specific angles and also administrate drugs - [50].

So the goal of this spin-off project, which is still under development, is to understand if the skin conductivity is able to clearly distinguish the SNS activity from PNS activity as the other signals measured by their devices can do. To verify it, a comparison under specific experimental conditions and procedures between GSR signal and the tilt table testing measures - already approved among the medical community - will be done and therefore report conclusions. For now, only a preliminary test can be presented which compares the skin conductivity measured by our sensor with the heart rate measured by their device. Being essentially a physical test, it consisted in seven stages which are described in the table 4.9 with the respective expected results. The experimental results of the three evaluated subjects are presented in 4.13, 4.14 and 4.15 and for each of them, the upper curve corresponds to the HR and the bottom curve is the SC. A remark for the picture 4.14 in which the upper curve is the RR interval (period) instead of HR, so it is the inverse curve.

Initial rest	HR ($\approx$ *constant*), SC ($\approx$ *constant*)
Exercise a hand	HR ($\uparrow$), SC ($\uparrow$)
Rest	HR ($\downarrow$), SC ($\downarrow$)
Abdominal exercise	HR ($\uparrow$), SC ($\uparrow$)
Rest	HR ($\downarrow$), SC ($\downarrow$)
Abdominal exercise	HR ($\uparrow$), SC ($\uparrow$)
Final rest	HR ($\downarrow$), SC ($\downarrow$)

Table 4.9: Experimental phases with expected results.

In these experiments, it is visible that, somehow, the SC accompanies the HR. Besides that, 4.14 and 4.15 clearly shows a match between the expected and experimental results - in 4.13 this is not so clear. Another interesting fact mostly observed in 4.14, is that the SC has a certain temporal delay, i. e., the time between the beginning of physical activity and the sweat glands activation, because the GSR sensor detects perspiration only on this last moment. The illustrated latency[7] is approximately 1 second and it can be an important parameter because, as already discussed with cardiologists, it would make sense to verify if it is a standard value. It is also noticed that the recovery time is shorter than the rising time, which allows to easily detect physical start-ups. Additionally, comparing the SC of figures 4.14 and 4.15 corresponding to a 23 years old men and a 40-50 years old men respectively, it can be observed that the *fight or flight* body response differs from person to person because, when they started the first abdominal exercise, the signal raised about $2\mu S$ for the younger subject and only about $1\mu S$ for the other. Besides this, it is also observed for the same moment that the skin conductivity rise time is longer for the older person.

Just to conclude this section, this experiment was only a starting point but the results are already promising because they indicate good indices of similarity between SC and HR signals which means that future examinations can be done. However, it must be referred two things: the first is concerned about the experimental protocol which still has to be clearly defined before any measurement and the second is that the synchronism between both (independent) measurement devices should be carefully done in further experiments not to loose the temporal reference of occurrences, because in this experiment, event registering was only done in the Android device. This negative issue is visible in the graphical results since there exists a temporal mismatch between the two biosignals. Nonetheless this does not affect the credibility of the conclusions made above.

[7]In this case, since it was used the Android application, this latency could be affected by other factors - see section 4.7

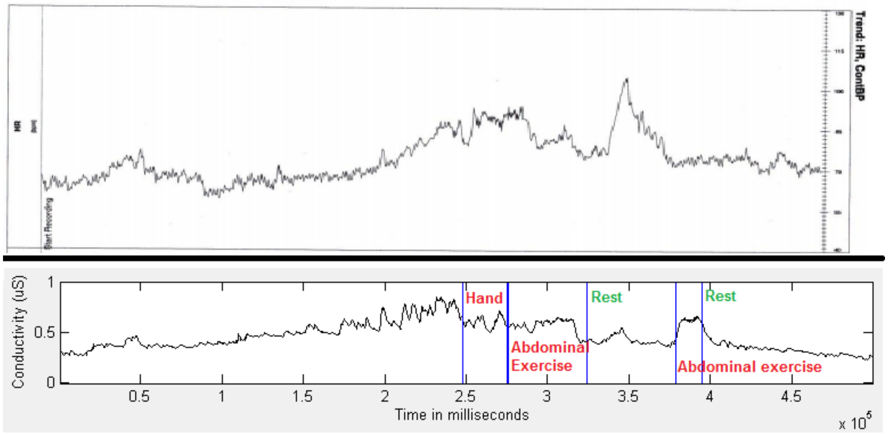


Figure 4.13: Tilt table context: results from a healthy female subject. Note: in this case no rest between hand and abdominal exercise was done.

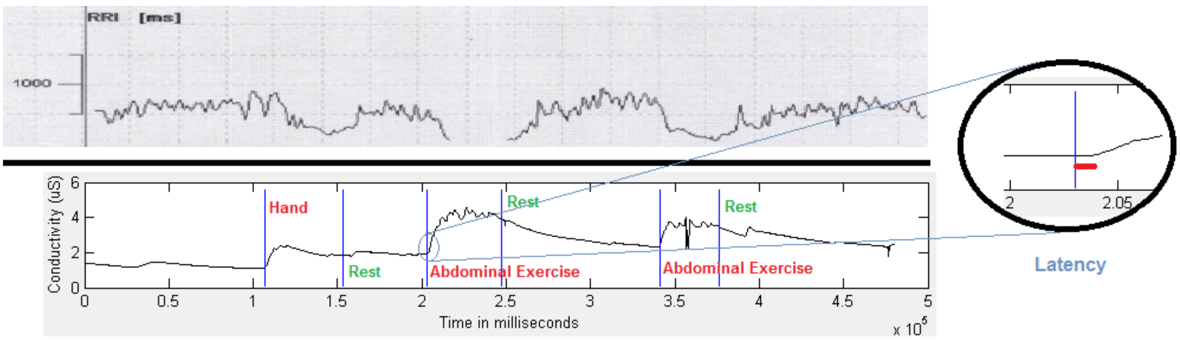


Figure 4.14: Tilt table context: results from a healthy masculine subject.

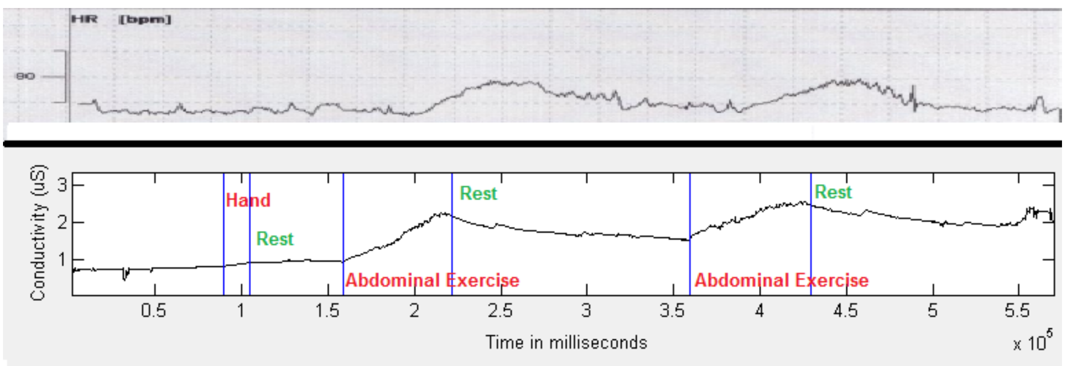


Figure 4.15: Tilt table context: results from a healthy masculine subject.

4.7 Remarks about the measured biosignals and registered occurrences

Throughout the experiments carried out, there were found some interesting details about the measured biosignals: SC and ECG. Some of these are described bellow:

- In general, it was observed that, somehow, the SC accompanies the HR.
- Not all the SC peaks are associated to an occurrence maybe due to the mental activity or the human delay when pressing a button.
- It was verified on a person with calloused hands, that the SC signal was very constant even exercising the hand where the electrodes were applied, which means that there was little release of perspiration.
- It was verified during the resting phase of a video game context that a peak with an amplitude of approximately $2.32\mu S$ and a rising time of $8.4s$ came up when the subject had itching in the back and scratched that body zone with the hand without the GSR sensor. In this situation, any changes in the HR was verified.
- The derivation of the respiration from the ECG signal maybe be done by several ways, and one of them is by tracking the amplitude of the R peaks [6]. Figure 4.16 was obtained from an experimental result. During inspiration the R amplitude decreases, and during the expiration it increases.

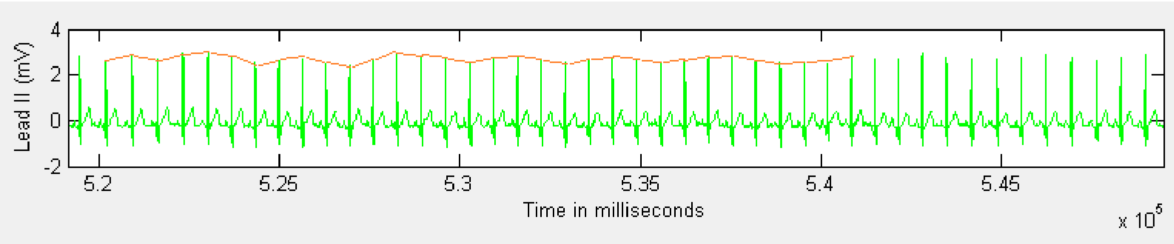


Figure 4.16: Respiration derived from the ECG.

The use of the GSR sensor in physical exercise tests has to be carefully managed since its working principle is related with perspiration. To conclude, it has to be referred that this platform was not built to measure the peak latency of the SC observed in the previous section, due to the way of registering occurrences, i. e. due the delay when pressing the button.

Chapter 5

Conclusions and future work

5.1 Summary

In this chapter is done a review of all the developed work and its conclusions. It is also pointed out some possible future work concerning the detection of human stress.

5.2 Conclusions

The human stress exists in all persons and it may come from several reasons such as tasks deadlines or anxiety. It is known that this stress can affect the human actions and therefore leading to undesired events such as accidents while driving a car. Another example of inherent stress consists in those persons who are severely impaired and have difficulties in accomplishing tasks as crossing a door by driving an electric wheelchair. So, the existence of a real-time stress detection system that could be adapted in these vehicular applications, will be very useful in the future in order to support the driving, for example to alert the driver about his/her stress state and to ask him/her to reduce the velocity, or in case of tetraplegic persons to implement a semi-automatic control of wheelchair when he/she is not able to accomplish some task.

However to build a complex system like this one, it is necessary to split it in several steps: discover how the stress can be measured, collect an increased amount of those measures, test and validate stress detection algorithms (and classifiers) and finally merge all the know-how into a full-fledged system. This dissertation was intended to serve as the basis for that system and it covered the two first steps.

In this work, it was presented a portable platform that acquires stress related signals as well as allows the registering of induced-stress occurrences with their respective timings. This platform uses two non-invasive wireless commercial sensors connected by Bluetooth to an Android based tablet. The comfortable and wearable sensors acquires the electrocardiogram and the skin conductivity signals, which carry reliable information about human stress according to the case study done in section 2.3. To receive data from sensors and register occurrences, it was developed an Android application that allows a person to interact with it in a friendly way. The registering of these occurrences can be done by voice or typing, and they will permit to identify, in an off-line analysing unit, which occurrences causes more stress. Furthermore, the application stores all acquired data and exhibits in real-time the SC and HR values. In section 4.2, some tests were performed in order to validate the adequate operation of the platform and it is concluded that all the requirements mentioned in section 3.2 are satisfied. Experimental results were obtained in different contexts such as vehicular navigation, video game and also in medical area, and therefore analysed off-line using a developed Matlab interface. This shows the multiple application of this platform.

Although, in general, the platform works well, it has some undesired problems such as the time to connect with both sensors. Sometimes, this time takes 1-3 seconds and other times it takes around 5 minutes, then, it would be desirable to reduce this waiting time. Concerning the memory problem related with the Matlab interface observed when the stored files contained $287MB$ of experimental data, a first step to try to fix it would be test it in a 64-bit version and in case that it works, it should be developed a single solution that requires that Matlab version.

5.3 Future work

The first steps to build the above mentioned stress detection system were mostly accomplished. To improve the ongoing of this project, some future work can be pointed out as is described bellow:

- Collect an increased amount of data, under the same experimental conditions, in order to analyse it via off-line. Therefore, stress algorithms can be validated and then adapted to the referred real-time operation.

- Improve the real-time HR algorithm since the final system has to work in real-time conditions. Also from the ECG, implement a real-time algorithm able to derive the respiration, which is another parameter related with the stress - data fusion.

- In the future, replace the commercial sensors by specifically developed ones which includes a low-power wireless connection, more adequate for the user and using a protocol that permits real-time operation.

- Concerning the actual Android application, include a feature that recognizes if the Android device lost the connection with the sensors due several reasons such as being distant from each other or battery died. In that case, warn the user and then do an adequate procedure.

- Include the Global Positioning System (GPS) feature in the Android application to provide the car location, which can be useful to detect dangerous points on the road.

To conclude, it is known that age plays an important role in the normal behaviour of the ANS, so it would be interesting to study the age influence in car driving situations since it requires quick reflexes from driver and normally they decrease with age.

Without forgetting the tilt table context (medical area), it has to be referred that this platform can be adapted to detect changes of the ANS in pathologies such as diabetic with a dysfunctional response. Another example is the analysing of the ANS in the absence of the nocturnal arterial tension (sleep) which can lead to an increased cardiovascular risk. Not proved yet, it is believed that this is due the ANS dysfunction.

Bibliography

[1] ISR (Institute of Systems and Robotics). Last access: March 2013. [Online]. Available: http://www.isr.uc.pt/

[2] Artificial Recognition System (ARS) model - Vienna University of Technology. Last access: March 2013. [Online]. Available: http://ars.ict.tuwien.ac.at/wiki/index.php/ARSModell

[3] A. Abdullah and U. Hassan, "Design and development of an emotional stress indicator (esi) kit," in *Sustainable Utilization and Development in Engineering and Technology (STUDENT), 2012 IEEE Conference on*, 2012, pp. 253–257.

[4] M. Strauss, C. Reynolds, S. Hughes, K. Park, G. Mcdarby, and R. W. Picard, "The handwave bluetooth skin conductance sensor," in *In ACII.* Springer, 2005, pp. 699–706.

[5] E. Vavrinsky, V. Stopjakova, and L. Majer, "Electrical biomonitoring towards mobile diagnostics of human stress influence," in *Applied Sciences in Biomedical and Communication Technologies, 2009. ISABEL 2009. 2nd International Symposium on*, 2009, pp. 1–6.

[6] E. Kaniusas, *Biomedical Signals and Sensors I: Linking Physiological Phenomena and Biosignals*, ser. Biological and Medical Physics, Biomedical Engineering. Springer, 2012.

[7] P. Karthikeyan, M. Murugappan, and S. Yaacob, "A review on stress inducement stimuli for assessing human stress using physiological signals," in *Signal Processing and its Applications (CSPA), 2011 IEEE 7th International Colloquium on*, 2011, pp. 420–425.

[8] M. T. Quazi and S. Mukhopadhyay, "Continuous monitoring of physiological parameters using smart sensors," in *Sensing Technology (ICST), 2011 Fifth International Conference on*, 2011, pp. 464–469.

[9] A. de Santos Sierra, C. Avila, G. Bailador del Pozo, and J. Guerra Casanova, "Stress detection by means of stress physiological template," in *Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on*, 2011, pp. 131–136.

[10] J. Zhai and A. Barreto, "Stress detection in computer users based on digital signal processing of noninvasive physiological variables," in *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, 2006, pp. 1355–1358.

[11] H. Patro, G. Raja, and S. Dandapat, "Classification of stressed speech using gaussian mixture model," in *INDICON, 2005 Annual IEEE*, 2005, pp. 342–346.

[12] J. Wagner, J. Kim, and E. Andre, "From physiological signals to emotions: Implementing and comparing selected methods for feature extraction and classification," in *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, 2005, pp. 940–943.

[13] J. Healey and R. Picard, "Detecting stress during real-world driving tasks using physiological sensors," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 6, no. 2, pp. 156–166, 2005.

[14] J. Mihalov, V. Stopjakova, R. Zalusky, and L. Majer, "Wireless measurement system for continuous biomonitoring towards stress diagnostics," in *Applied Electronics (AE), 2012 International Conference on*, 2012, pp. 183–186.

[15] Shimmer-Research. Last access: March 2013. [Online]. Available: http://www.shimmer-research.com/

[16] MIT Media Lab - affectiva. Last access: March 2013. [Online]. Available: http://www.affectiva.com/q-sensor/features/

[17] Bio-Medical Instruments, Inc. Last access: March 2013. [Online]. Available: http://bio-medical.com/

[18] India Mart. Last access: March 2013. [Online]. Available: http://www.indiamart.com/

[19] BioPac Systems. Last access: March 2013. [Online]. Available: http://www.biopac.com/

[20] Zephyr. Last access: March 2013. [Online]. Available: http://www.zephyr-technology.com/products/

[21] Polar. Last access: March 2013. [Online]. Available: http://www.polar.com/en/products/accessories?product=mobile

[22] Quirumed. Last access: March 2013. [Online]. Available: http://www.quirumed.com/pt/

[23] Ec21. Last access: March 2013. [Online]. Available: http://www.ec21.com/product-details/Heart-Rate-Monitor--7357073.html

[24] Matlab. Last access: May 2013. [Online]. Available: http://www.mathworks.com/products/matlab/

[25] Dropbox. Last access: April 2013. [Online]. Available: https://www.dropbox.com

[26] *Shimmer Packet Format - Bluetooth Boilerplate.*

[27] *GSR Shimmer sensor - User Guide rev 1.5.*

[28] *ECG Shimmer sensor - User Guide rev 1.4.*

[29] Realterm - serial capture program. Last access: March 2013. [Online]. Available: http://realterm.sourceforge.net/

[30] ECG sensor from Schimmer-Research. Last access: March 2013. [Online]. Available: http://www.shimmer-research.com/wp-content/uploads/wpsc/product_images/Shimmer-ECG-Module-Full-Enc.jpg

[31] Asus Nexus 7 - tablet. Last access: March 2013. [Online]. Available: http://www.google.com/nexus/7/specs/

[32] Platform versions distribution. Last access: April 2013. [Online]. Available: http://developer.android.com/about/dashboards/index.html#Platform

[33] Android Developer Tools. Last access: March 2013. [Online]. Available: http://developer.android.com/sdk/index.html

[34] Eclipse Platform - Software tool. Last access: March 2013. [Online]. Available: http://www.eclipse.org/

[35] Android Tablet Asus Nexus 7. Last access: March 2013. [Online]. Available: http://www.google.com/nexus/7/

[36] Android for developers - Activities. Last access: April 2013. [Online]. Available: http://developer.android.com/guide/components/activities.html

[37] Android for developers - Activity life-cycle. Last access: April 2013. [Online]. Available: http://developer.android.com/training/basics/activity-lifecycle/starting.html#launching-activity

[38] Android for developers - Bluetooth. Last access: April 2013. [Online]. Available: http://developer.android.com/guide/topics/connectivity/bluetooth.html

[39] Android for developers - Asynctask. Last access: April 2013. [Online]. Available: http://developer.android.com/reference/android/os/AsyncTask.html

[40] Android for developers - process and threads. Last access: April 2013. [Online]. Available: http://developer.android.com/guide/components/processes-and-threads.html#Threads

[41] Android for developers - thread. Last access: April 2013. [Online]. Available: http://developer.android.com/reference/java/lang/Thread.html

[42] Networkonmainthreadexception. Last access: April 2013. [Online]. Available: http://developer.android.com/reference/android/os/NetworkOnMainThreadException.html

[43] Android for developers - external storage directory. Last access: April 2013. [Online]. Available: http://developer.android.com/reference/android/os/Environment.html#getExternalStorageDirectory()

[44] Android for developers - Recognizer Intent. Last access: April 2013. [Online]. Available: http://developer.android.com/reference/android/speech/RecognizerIntent.html

[45] Dropbox for developers - Dropbox Core Api. Last access: April 2013. [Online]. Available: https://www.dropbox.com/developers/core

[46] AChartEngine - charting software library for Android applications. Last access: April 2013. [Online]. Available: http://www.achartengine.org/

[47] Android for developers - handler. Last access: April 2013. [Online]. Available: https://developer.android.com/training/multiple-threads/communicate-ui.html

[48] The BioSig Project - biomedical signal processing. Last access: April 2013. [Online]. Available: http://biosig.sourceforge.net/

[49] Ledalab - skin conductance signal processing. Last access: April 2013. [Online]. Available: http://www.ledalab.de/

[50] T. Schwalm, *Modern Tilt Table Testing and Non-Invasive Monitoring.* Abw Wissenschaftsverlag, 2006, last access: May 2013. [Online]. Available: http://books.google.pt/books?id=pirfAAAACAAJ

[51] Android for developers. Last access: April 2013. [Online]. Available: http://developer.android.com/

[52] M. Murphy, *Beginning Android 2*, ser. Apresspod Series. Apress, 2010, last access: May 2013. [Online]. Available: http://books.google.pt/books?id=2XeNswkT_2YC